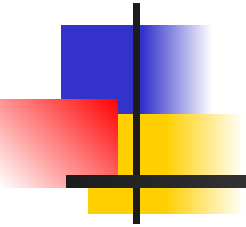


Organización de Computadoras



Clase 4



Temas de clase

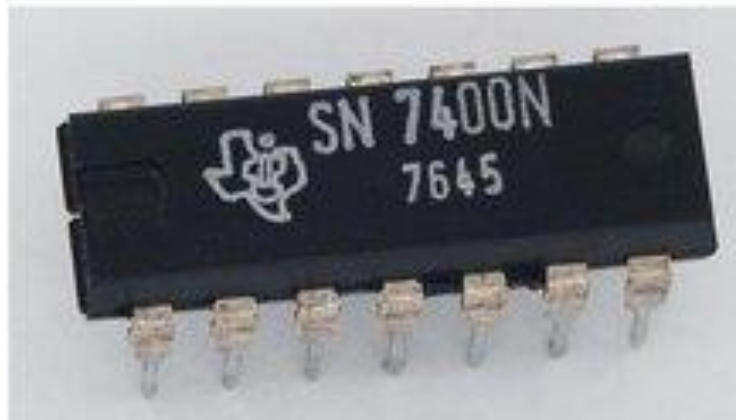
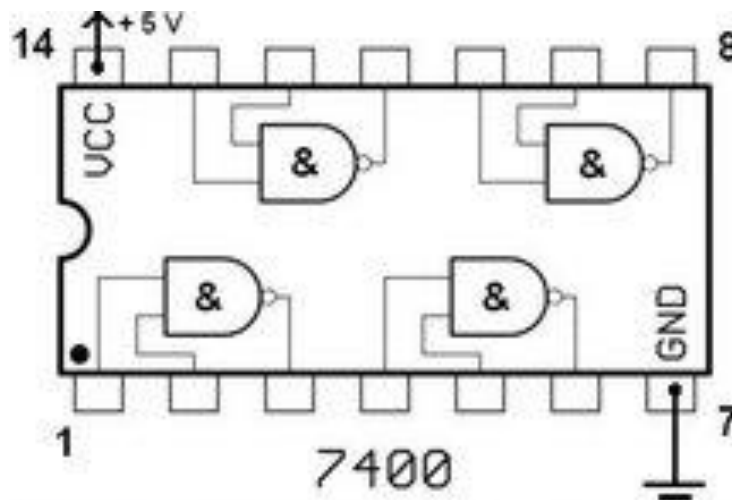
- Circuitos Lógicos Combinacionales
- Circuitos Lógicos Secuenciales



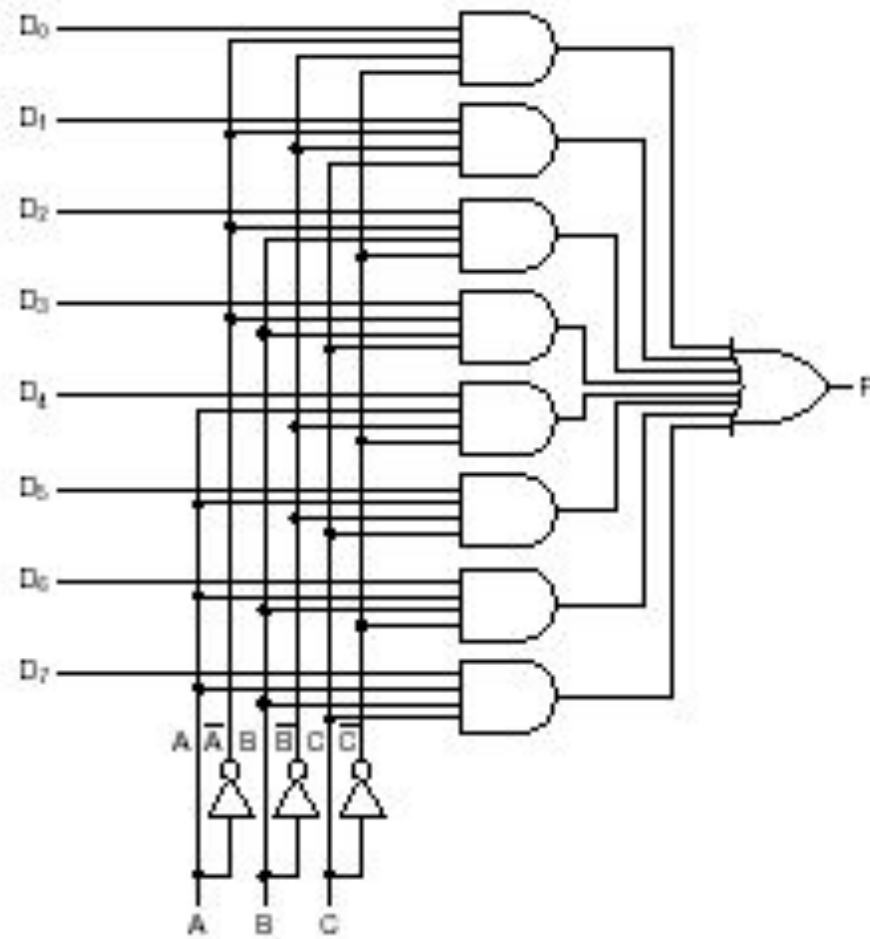
Circuitos Combinacionales o Combinatorios

- Responden a los valores lógicos en las entradas, la salida está determinada exclusivamente por los valores de las entradas en ese instante.
- Si cambia la entrada, cambia la salida.
- Los valores pasados de las entradas no influyen en los valores de las salidas.

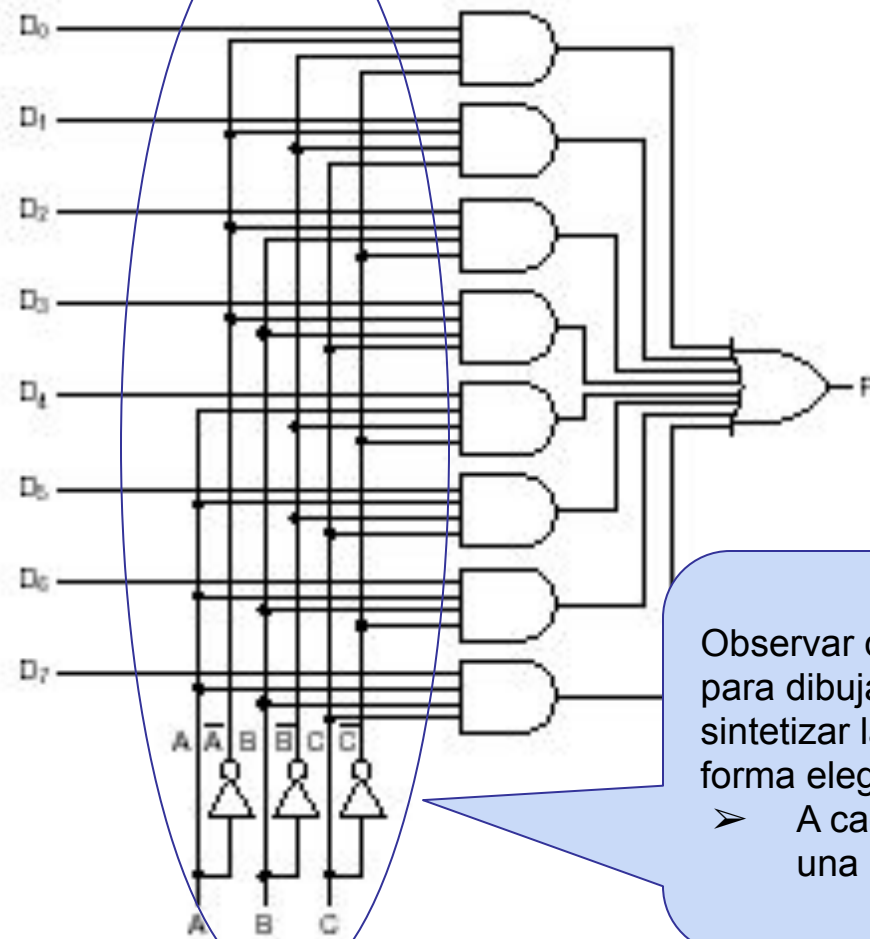
Puertas lógicas en un chip



Ejemplo 1



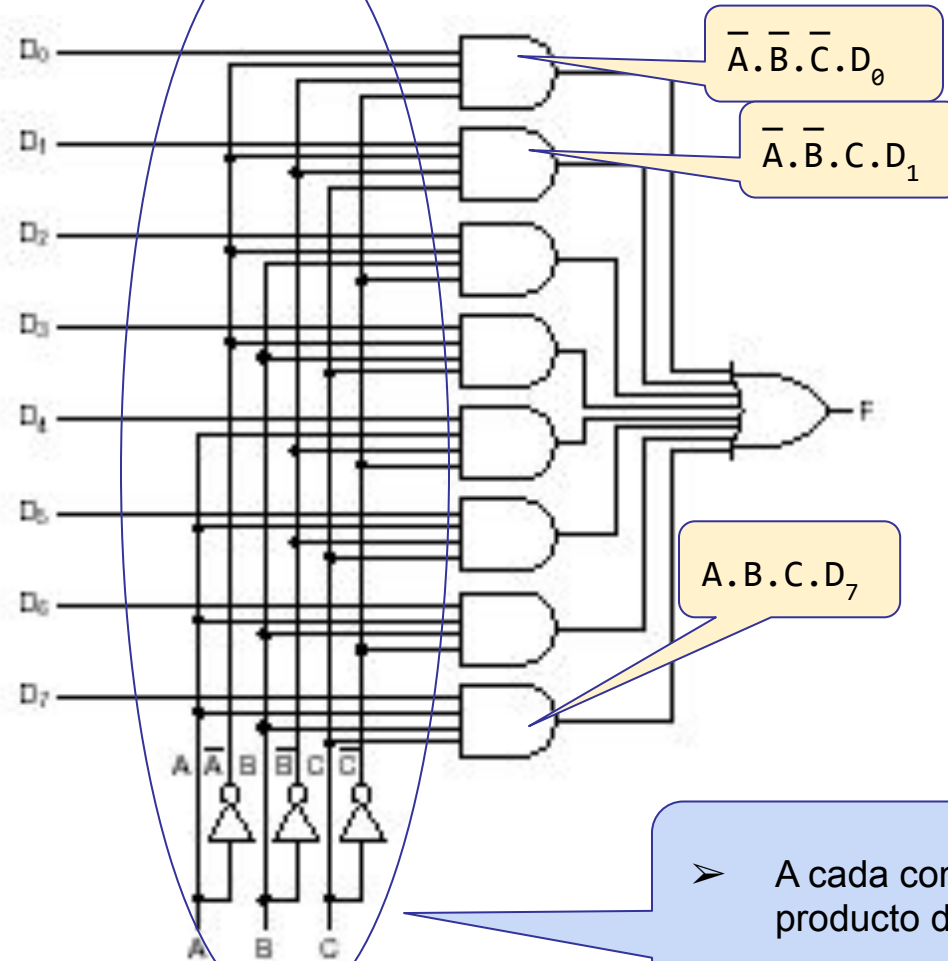
Ejemplo 1



Observar que es similar a lo que hicimos para dibujar el circuito de clase 1 para sintetizar la función mayoría. Es una forma elegante y ordenada de dibujar.

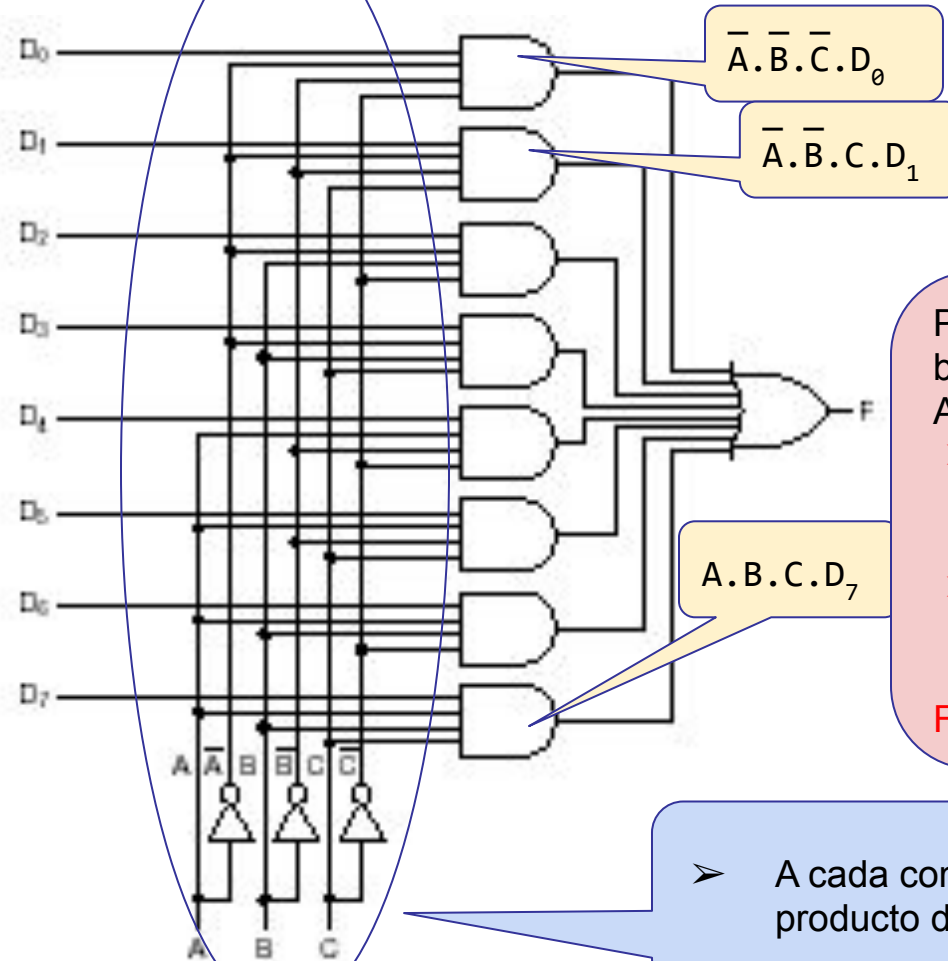
- A cada compuerta AND le llega un producto diferente de A, B y C

Ejemplo 1



➤ A cada compuerta AND le llega un producto diferente de A,B y C

Ejemplo 1



Para cada combinación binaria de entrada en A,B y C,

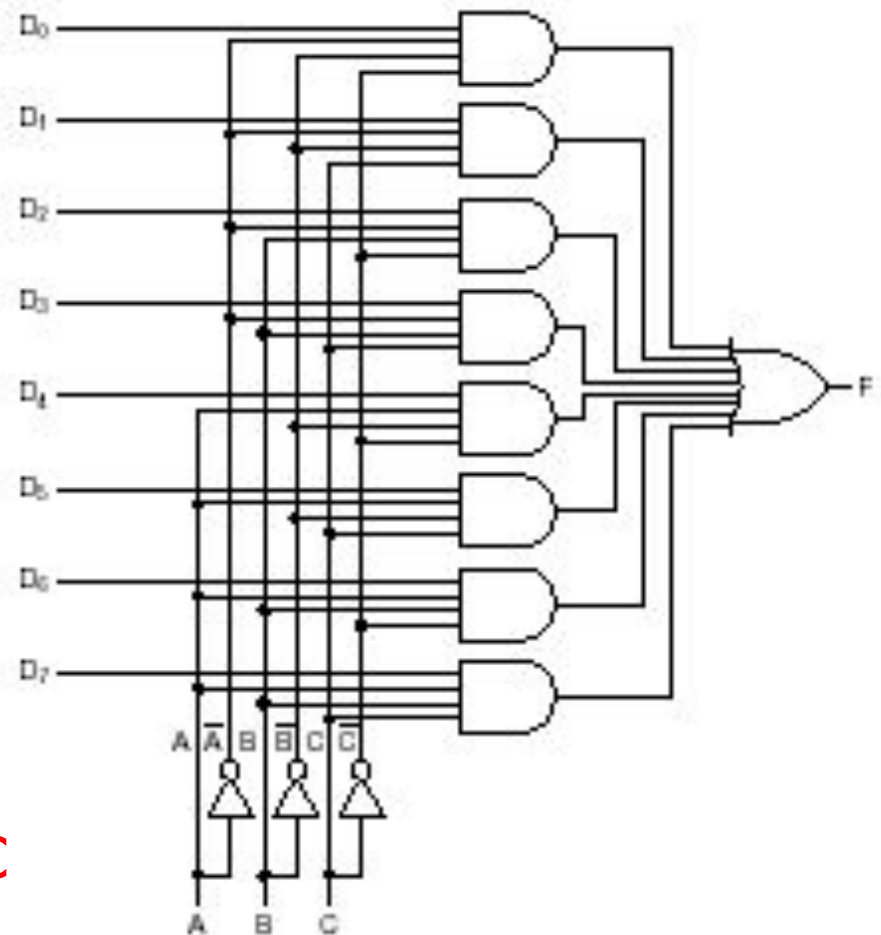
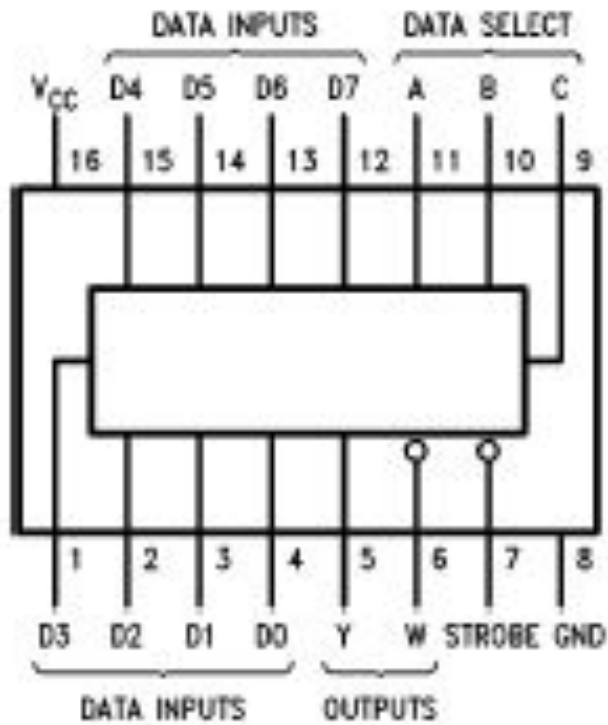
- 7 de las 8 AND entregarán 0 en su salida
- Una entregará el valor D_i

F tendrá el valor D_i

- A cada compuerta AND le llega una producto diferente de A,B y C

Ejemplo 1

Multiplexor de 8 entradas •74151

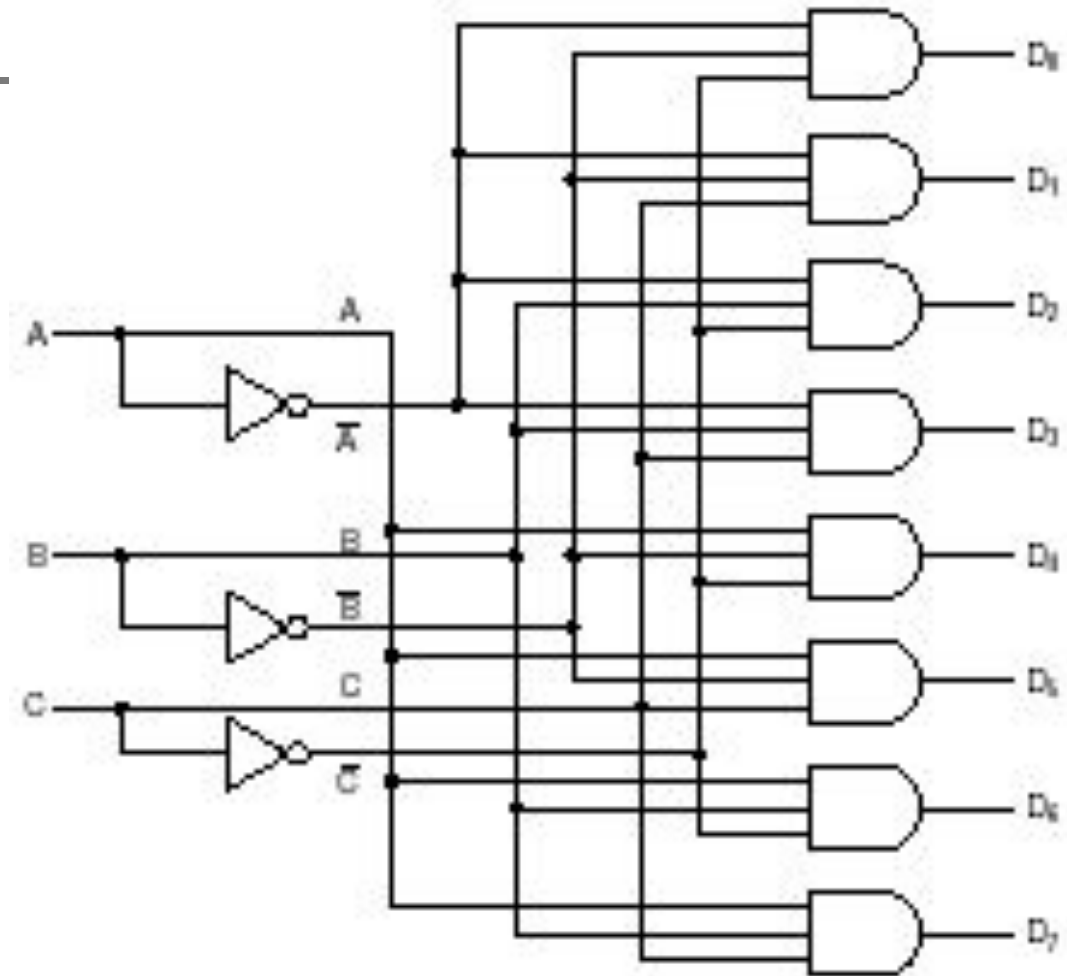


Según valor de entradas A, B y C
 $F = D_x$

Ejemplo 2

Para cada combinación de las entradas A, B y C sólo UNA de las salidas D_x vale '1'

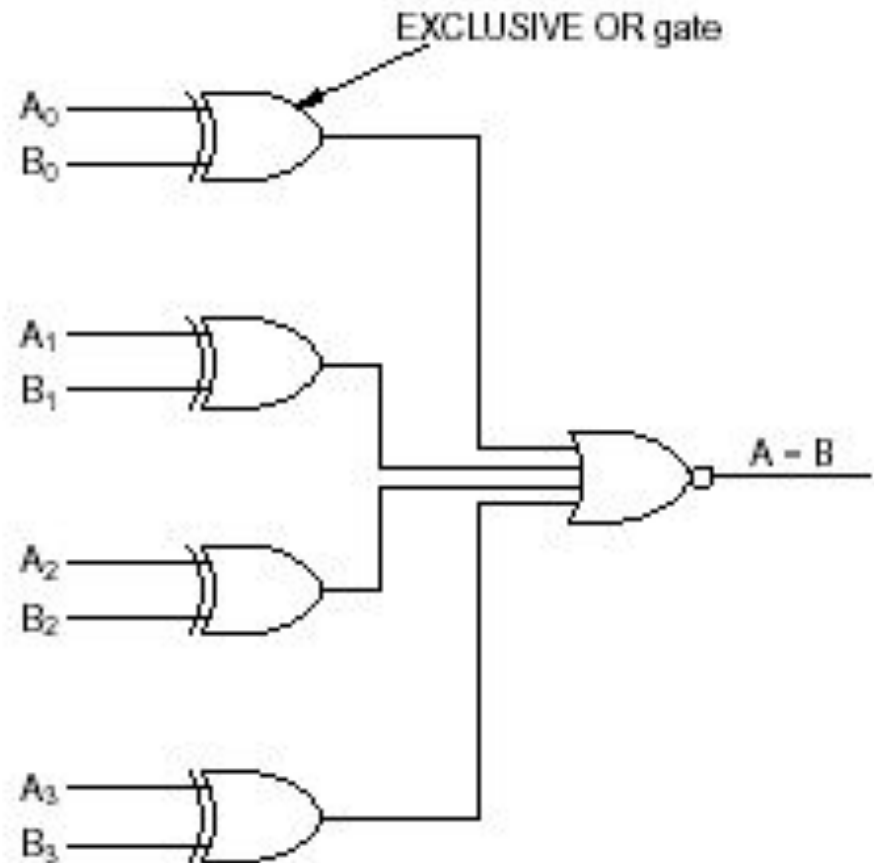
Decodificador 3 a 8



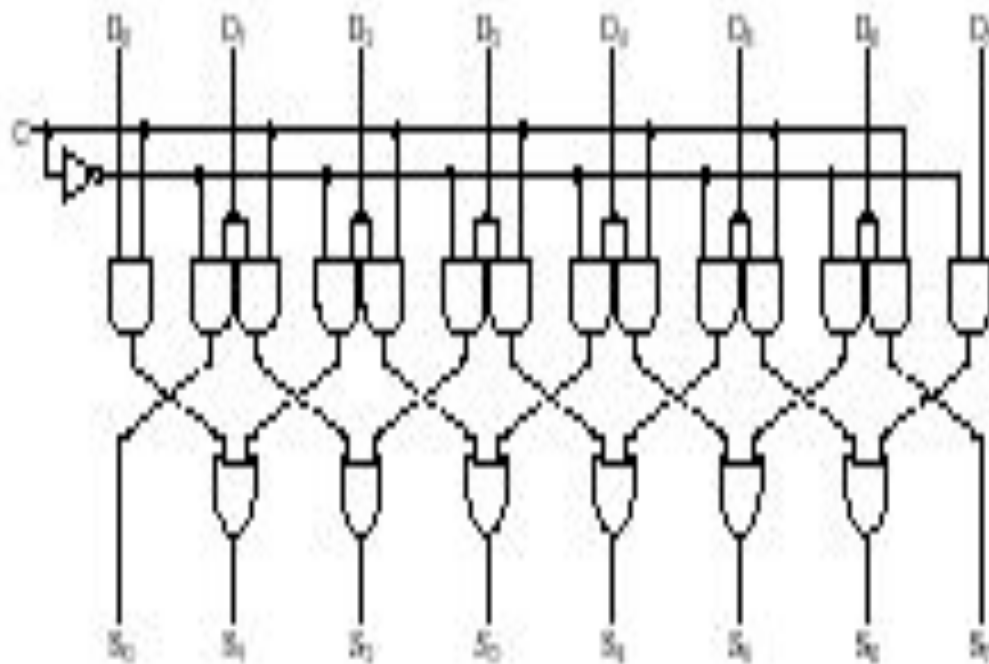
Ejemplo 3

Si todos los bits A_i son iguales a los B_i , la salida es '1'

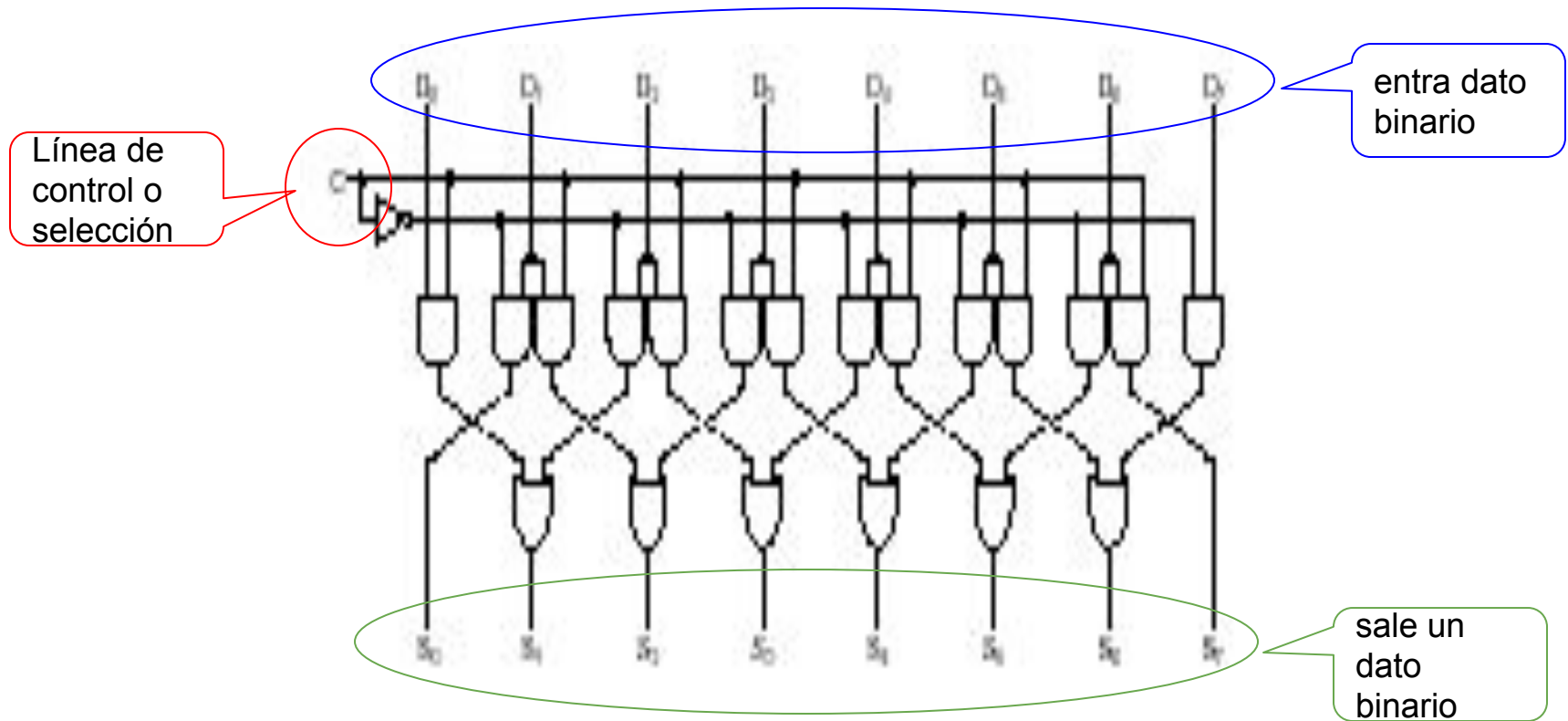
Comparador de 4 bits



Ejemplo 4

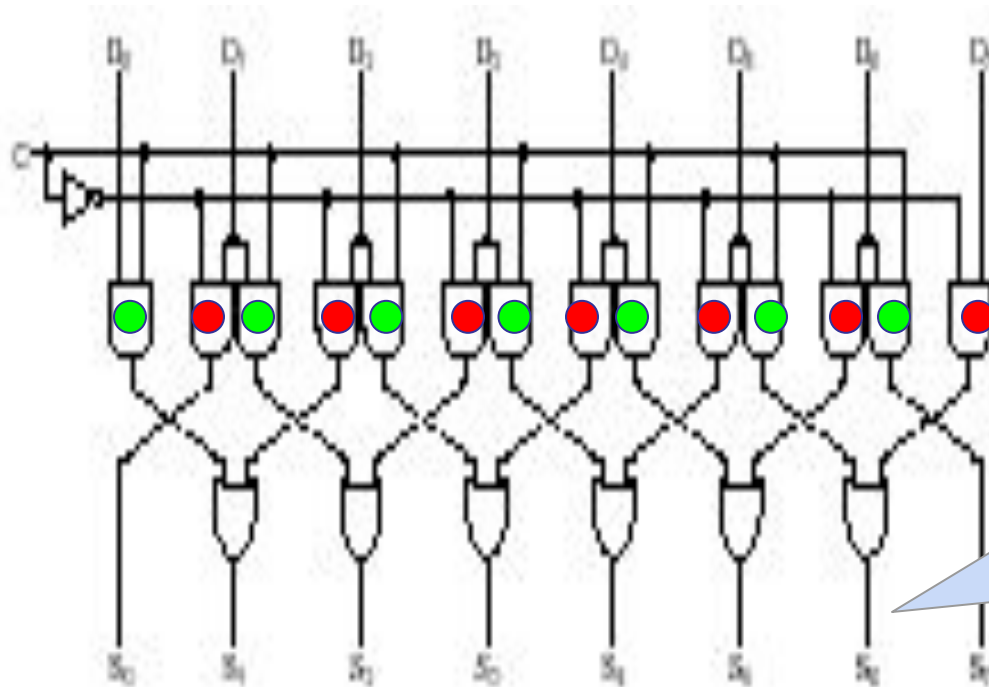


Ejemplo 4



"dejan pasar" el dato D_i
($1.D_i = D_i$)

AND izquierdas

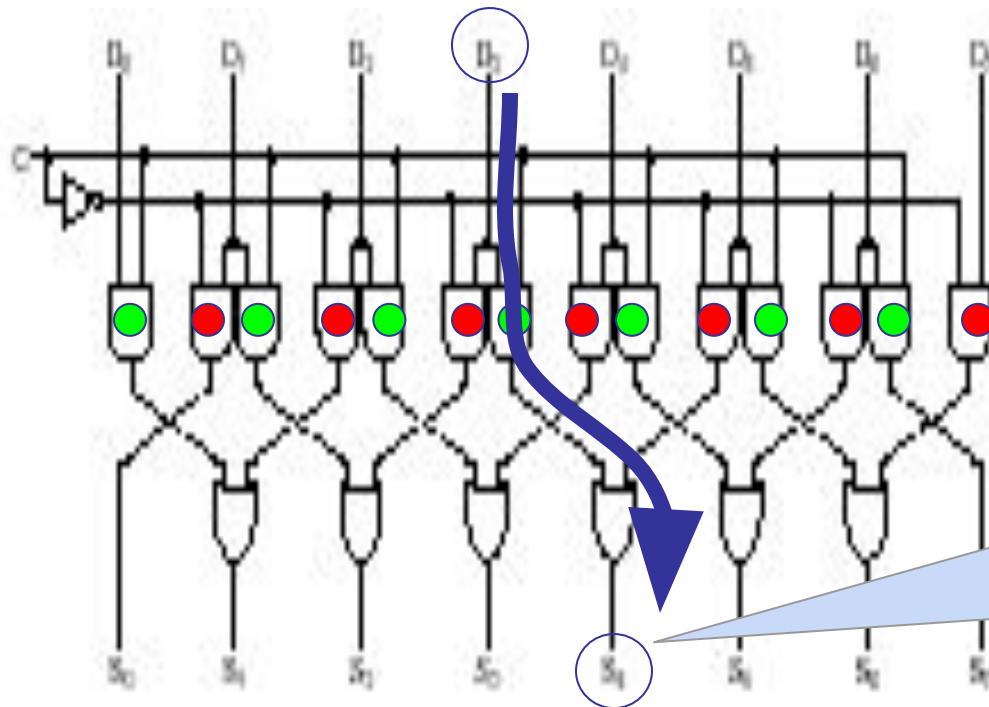


Su salida dependerá
del valor D_i ya que
 $0 + D_i = D_i$

Ejemplo 4

Si **C=1**
AND derechas
"dejan pasar" el
dato D_i
($1 \cdot D_i = D_i$)

AND izquierdas
tienen 0 a la
salidas sin
importar D_i
($0 \cdot D_i = 0$)



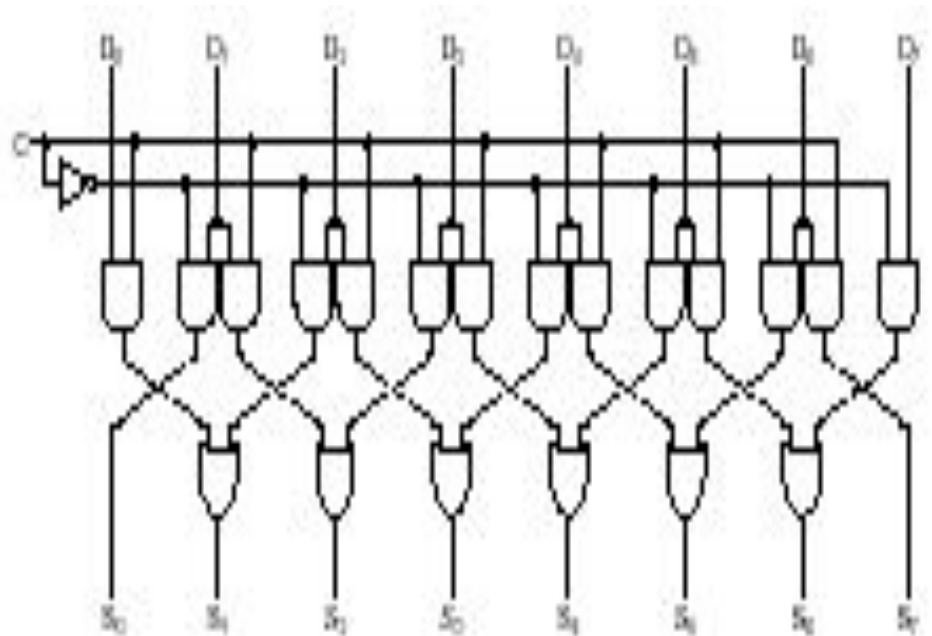
Cada OR recibe un
0 de una **AND** y un
 D_i de la otra **AND**

Su salida dependerá
del valor D_i
($D_i + 0 = D_i$)

Ejemplo 4

Desplazador de 1 bit

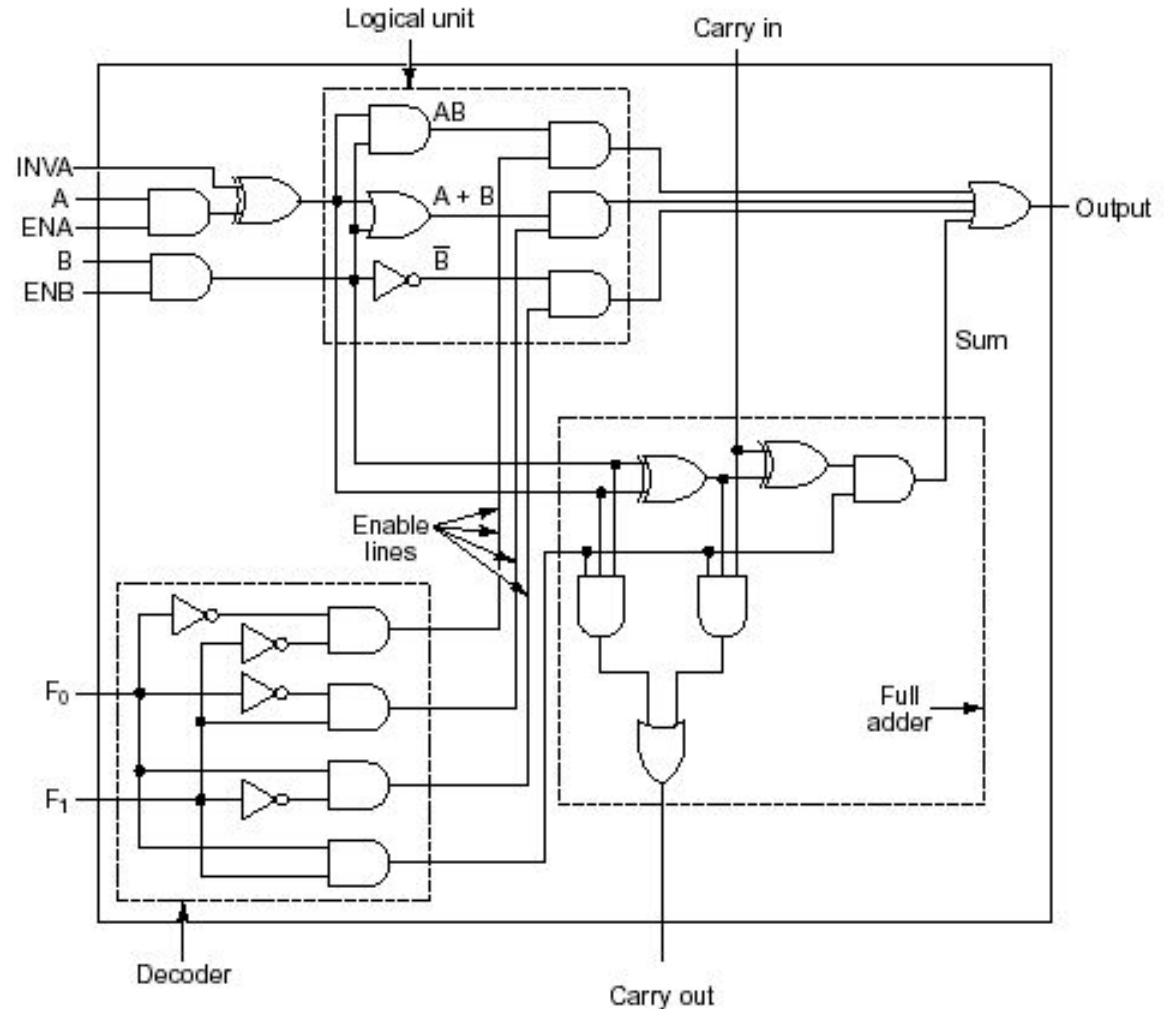
Según el valor de la entrada C se 'correrán' un lugar a derecha o izquierda.



Ejemplo 5

1 bit de ALU

Según $F_1 F_0$
será la función
que se realizará
sobre A y B.

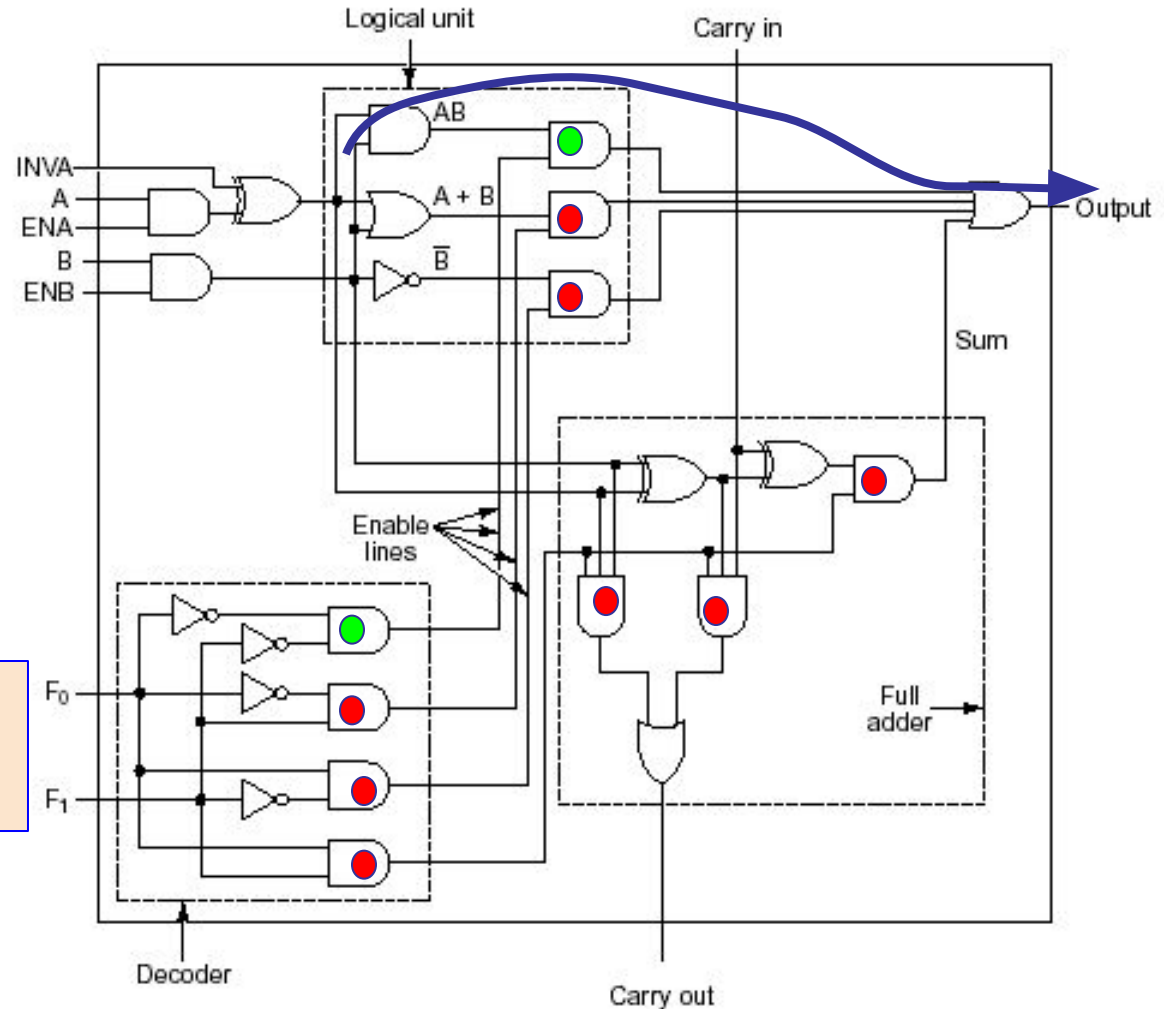


Ejemplo 5

1 bit de ALU

Según $F_1 F_0$
será la función
que se realizará
sobre A y B.

0
0

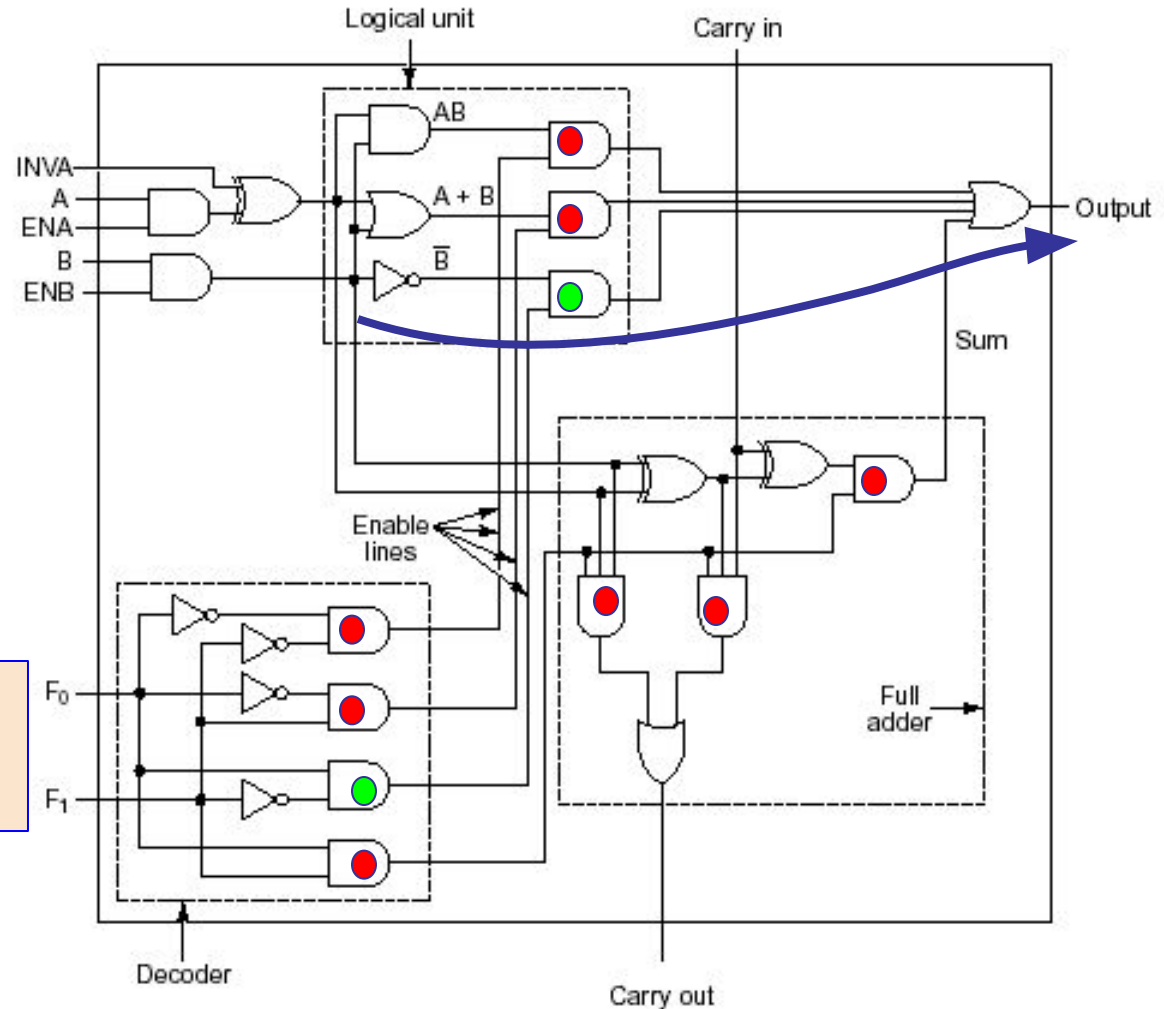


Ejemplo 5

1 bit de ALU

Según $F_1 F_0$
será la función
que se realizará
sobre A y B.

1
0

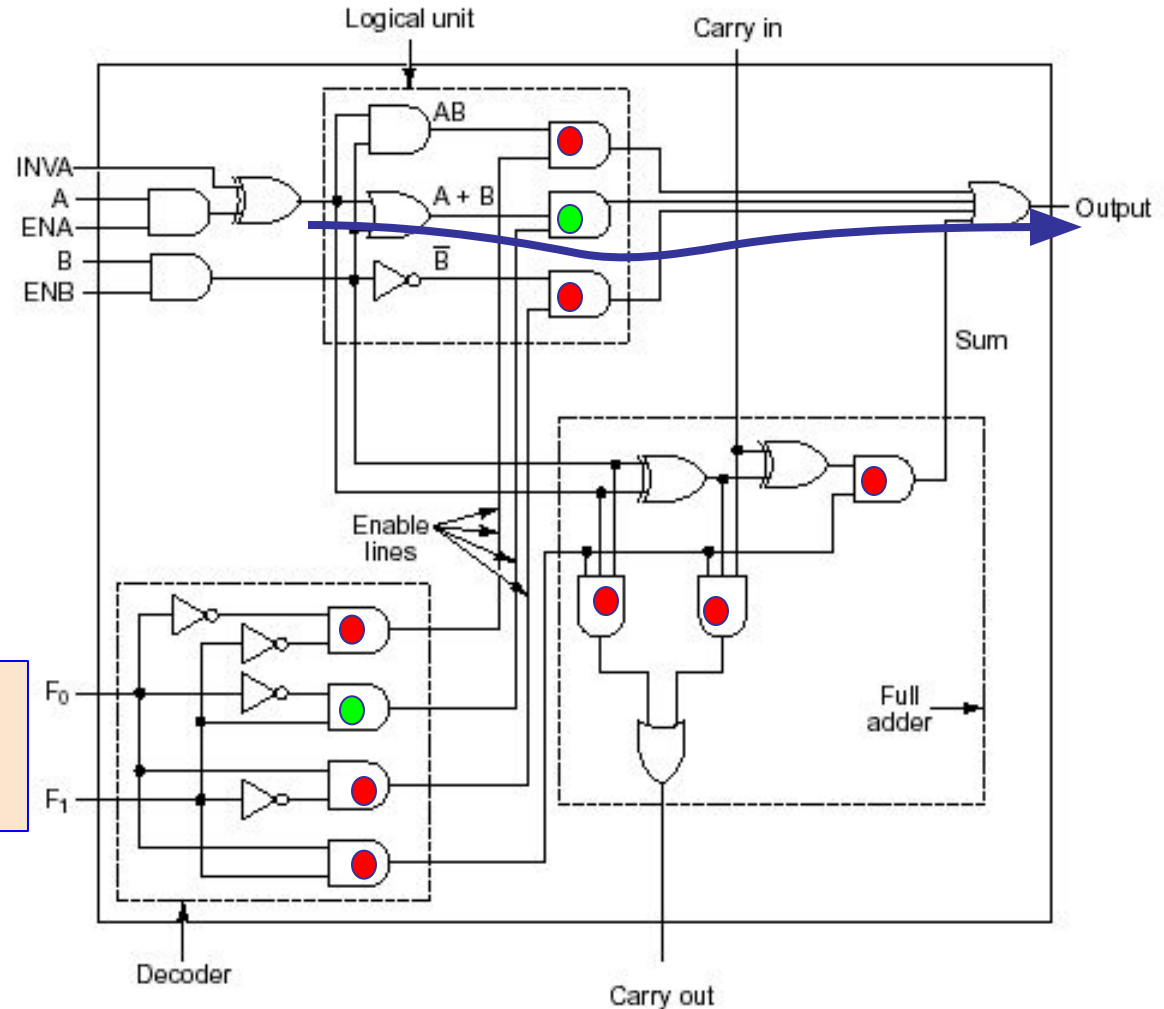


Ejemplo 5

1 bit de ALU

Según $F_1 F_0$
será la función
que se realizará
sobre A y B.

0
1

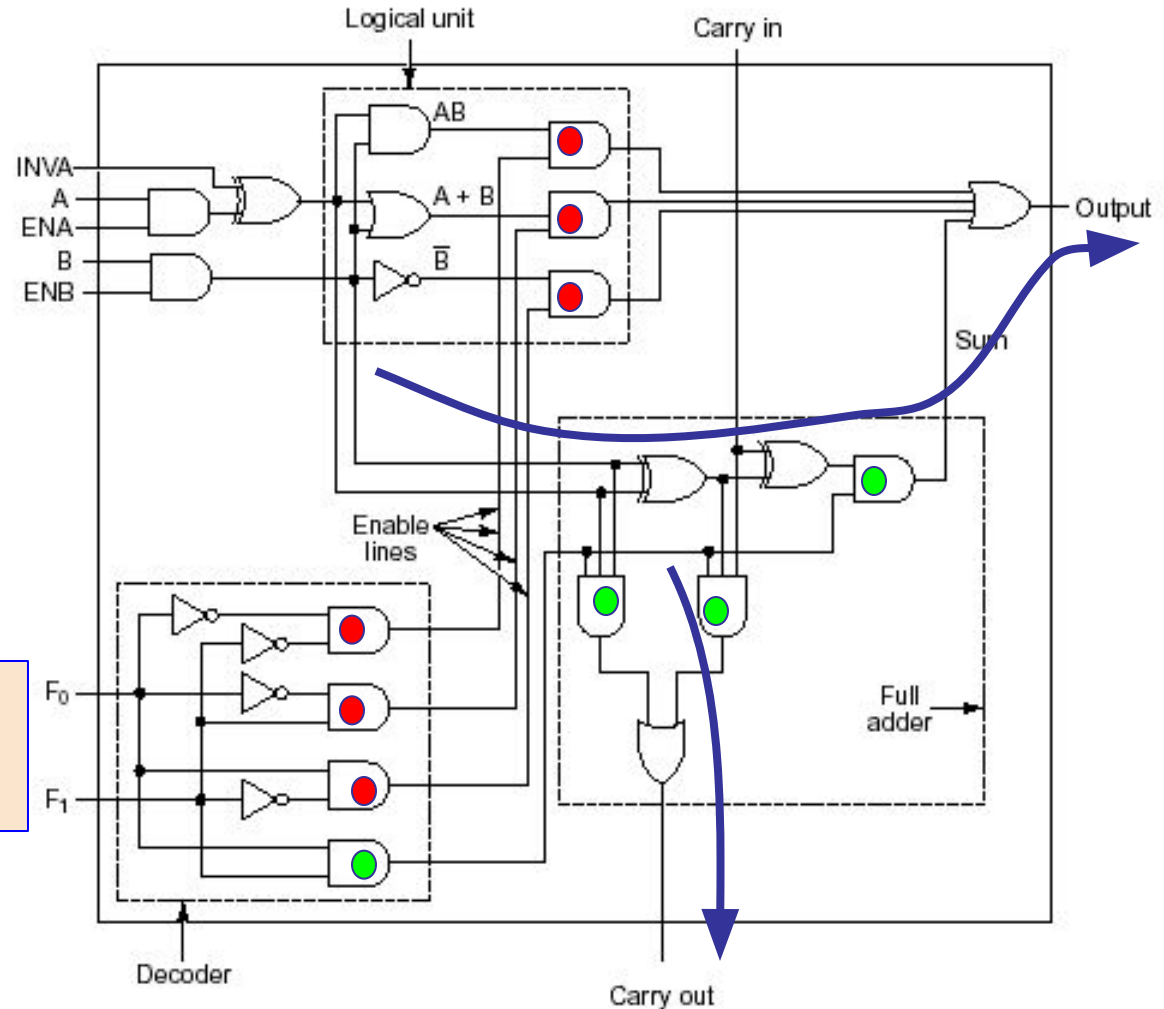


Ejemplo 5

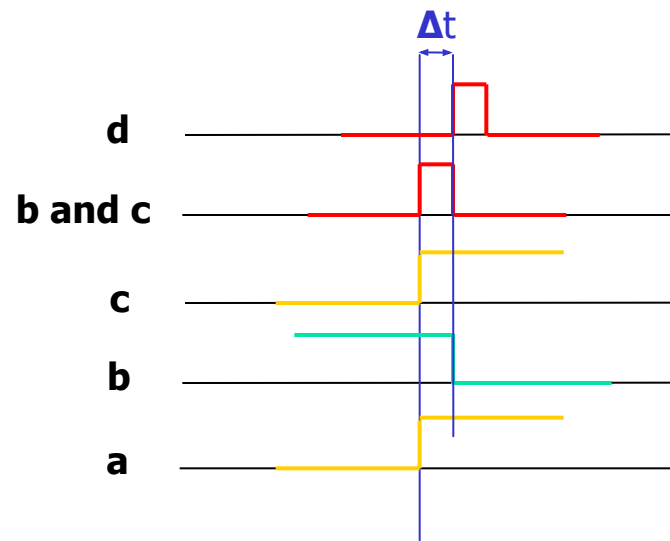
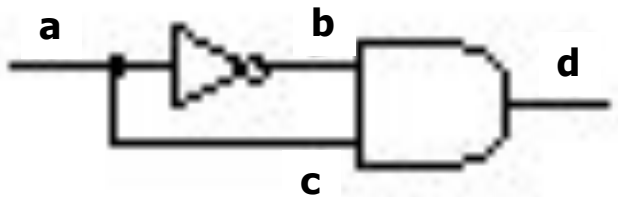
1 bit de ALU

Según $F_1 F_0$
será la función
que se realizará
sobre A y B.

1
1



Respuesta temporal



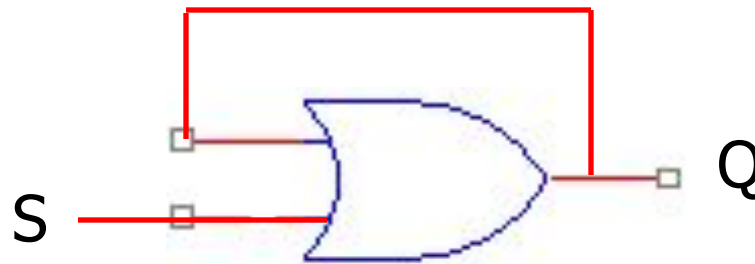
Suponemos que los retardos de compuerta Δt son iguales



Circuitos Secuenciales

- Las salidas dependen tanto de las entradas como del estado interno del circuito.
 - ¿Qué es el estado interno del circuito?
- Tienen la característica de “almacenar” valores lógicos internamente.
- Estos valores se almacenan aunque las entradas no estén.

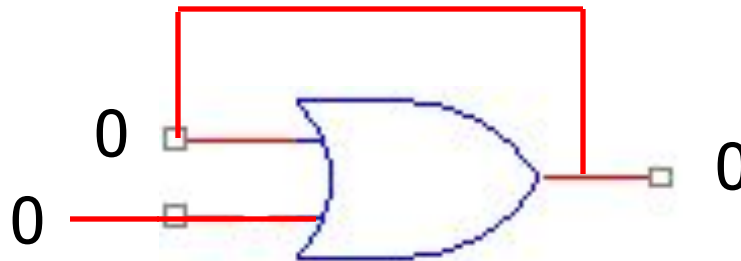
¿Cómo se almacena un valor lógico?



- La salida es también entrada
- En ningún circuito combinatorio una salida transportaba información hacia la entrada
- La ecuación lógica

$$Q = Q + S$$

¿Cómo se ...?(2)



➤ Supongamos que $S=0$ y $Q=0$

$$Q = Q + S = 0 + 0 = 0$$

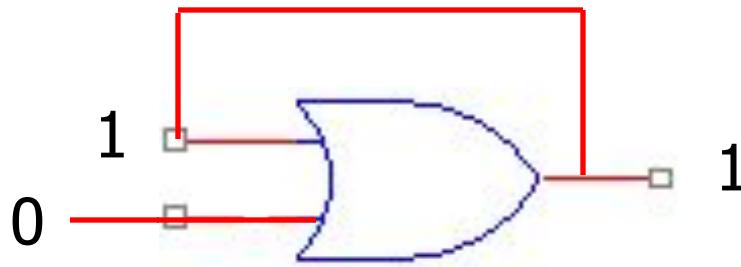
¿Cómo se ...?(3)



➤ Ahora $S=1$

$$Q = Q + S = 1 + 1 = 1$$

¿Cómo se ...?(4)

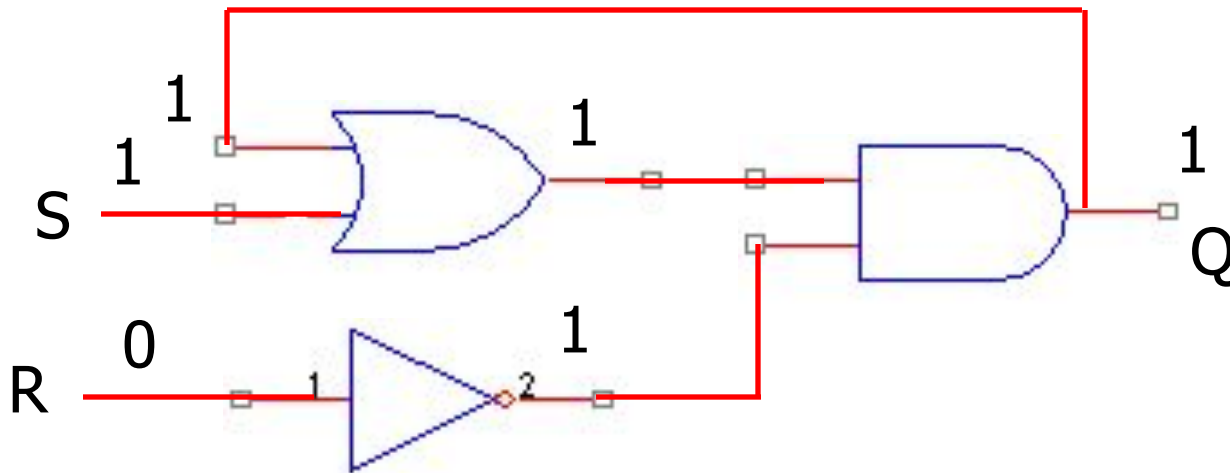


➤ Ahora $S=0$

$$Q=Q+S=1+0=1$$

➤ Una vez que la salida Q toma el valor 1 no hay forma de volver a 0

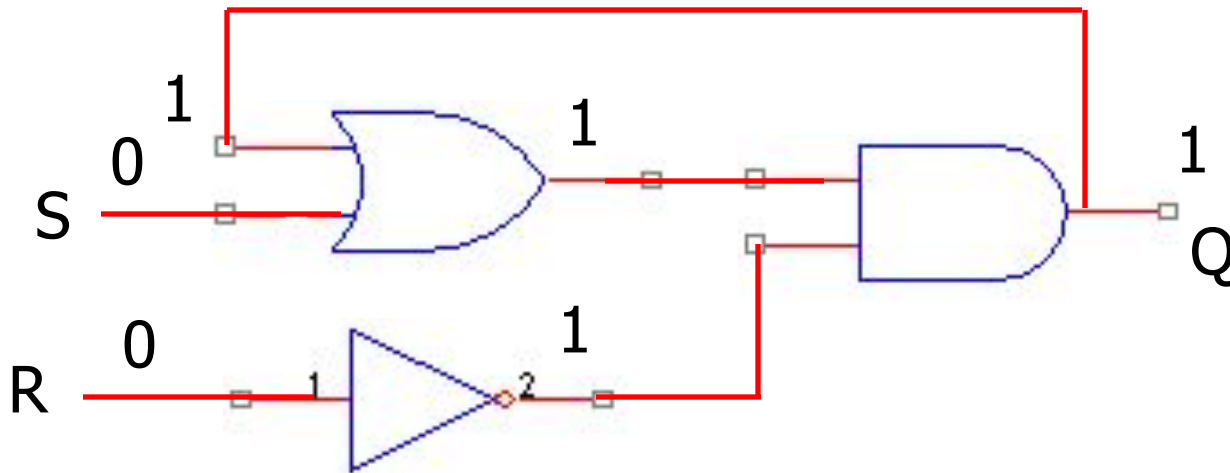
¿Cómo se ...?(5)



Ahora $S=1$ y $R=0$, $Q=1$

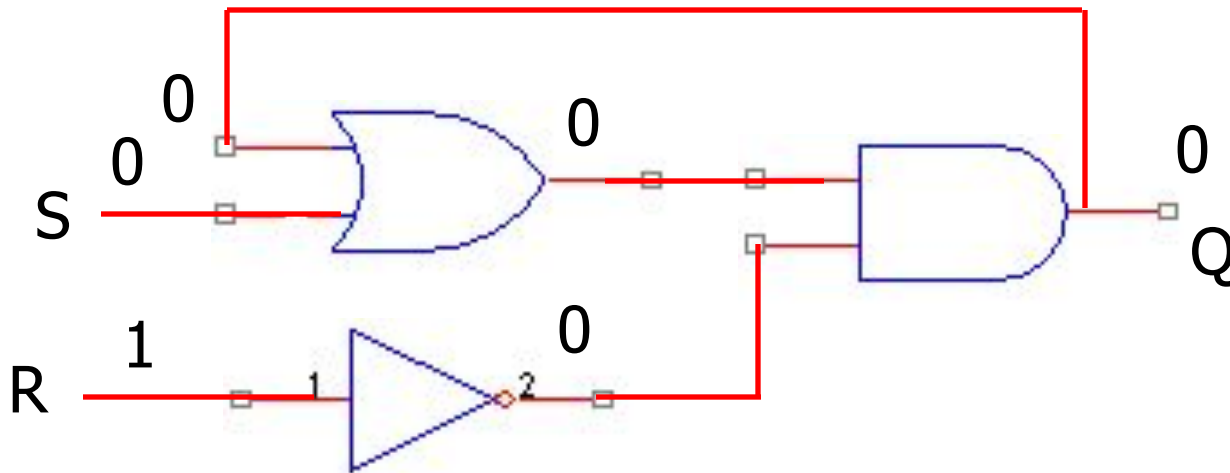
$$Q = (Q + S) \cdot \bar{R}$$

¿Cómo se ...?(6)



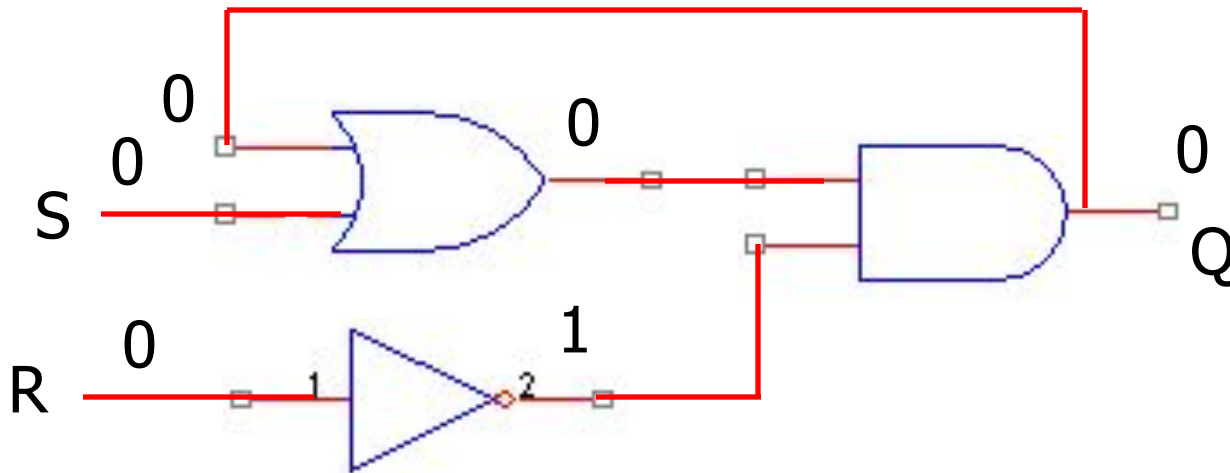
➤ Si ahora $S=0$ y $R=0$, $Q=1$. Nada cambia.

¿Cómo se ...?(7)



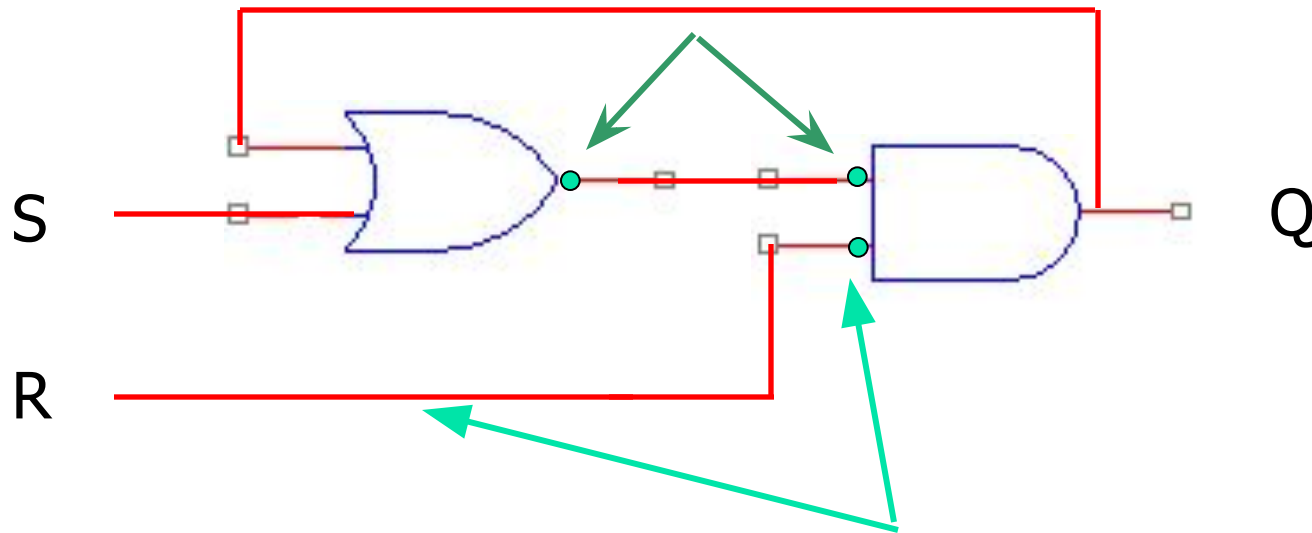
➤ Si ahora $S=0$ y $R=1$, $Q=0$.

¿Cómo se ...?(8)

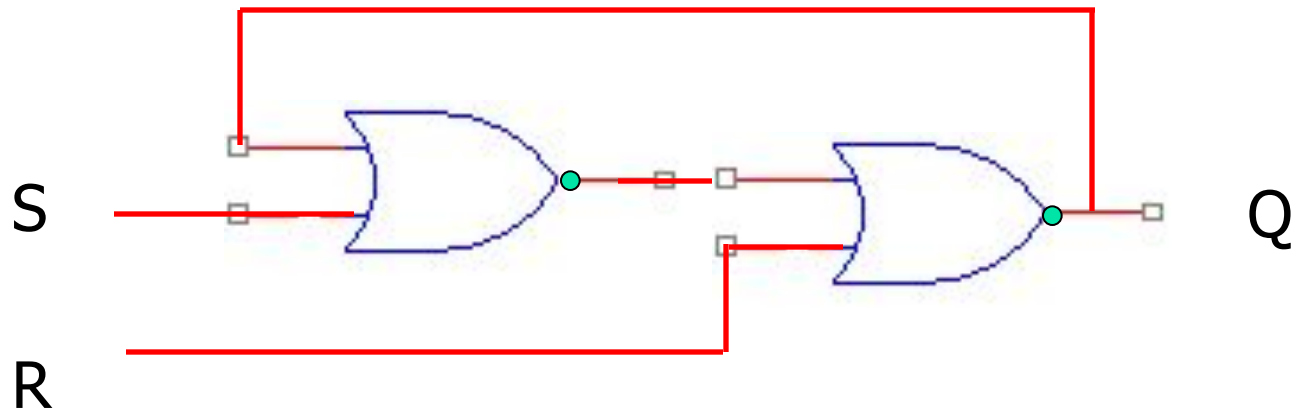


- Si ahora $S=0$ y $R=0$, $Q=0$.
- S puede cambiar y se reflejará en Q

¿Cómo se ...?(9)

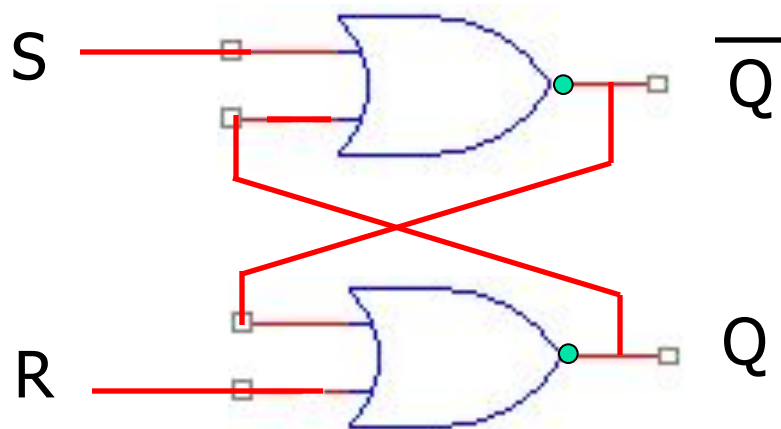


¿Cómo se ...?(10)



❖ Finalmente queda así

FLIP-FLOP SR



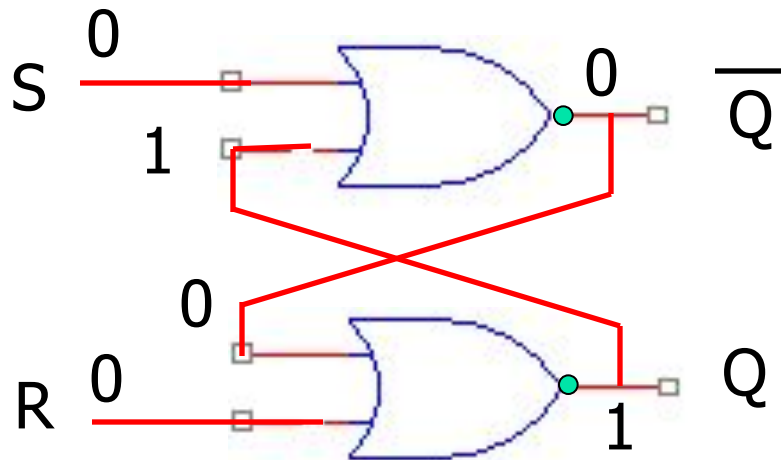
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Prohibido



FLIP-FLOP SR(2)

- Aparece la salida Q_{n+1}
- Q_n = salida anterior
- S = Set = poner a 1
- R = Reset = poner a 0
- Las salidas Q y \overline{Q} son complementarias

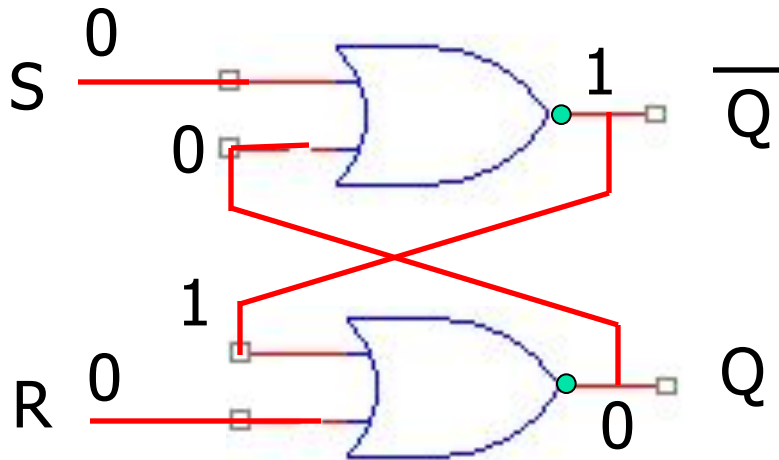
FLIP-FLOP SR(3)



Supongamos S y $R = 0$ y $Q = 1$

➤ Mientras que S y $R = 0$, Q permanece en 1

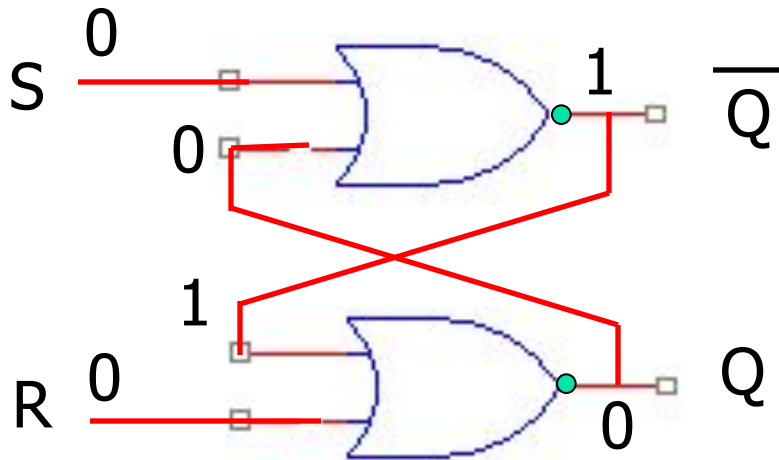
FLIP-FLOP SR(4)



Supongamos S y $R = 0$ y $Q = 0$

➤ Mientras que S y $R = 0$, Q permanece en 0

FLIP-FLOP SR(4)



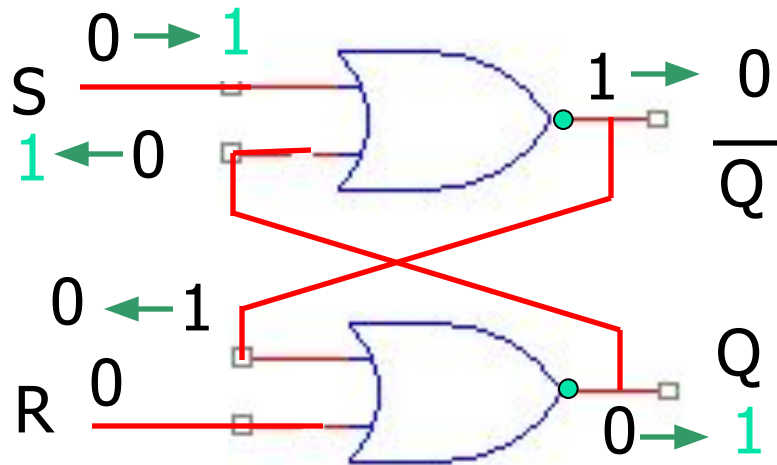
Si S y R = 0,

➤ “recuerda” cual era el estado anterior.

Supongamos S y R = 0 y Q = 0

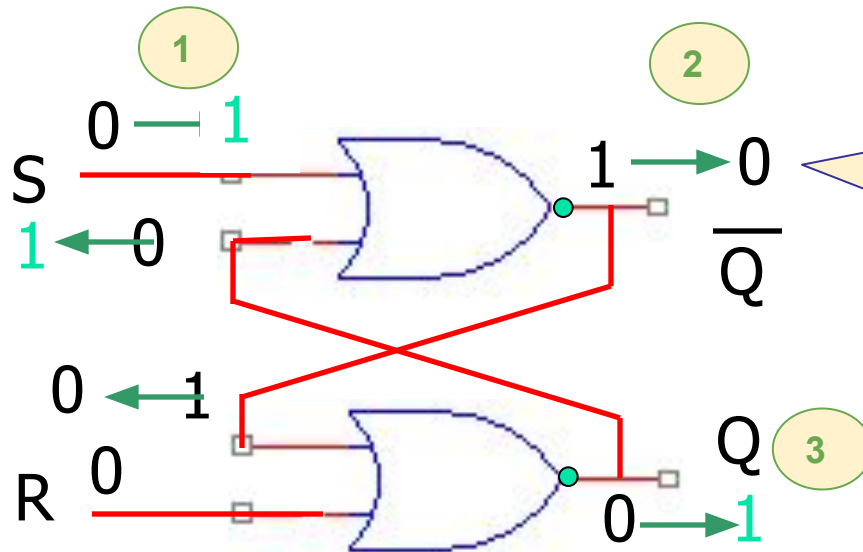
➤ Mientras que S y R = 0, Q permanece en 0

FLIP-FLOP SR(5)



❖ Si ahora pasamos de S y R=0 a S=1 y R=0

FLIP-FLOP SR(5)



Observar el orden de los cambios

Tarda 2 tiempos de compuerta en llegar al estado final esperado.

❖ Si ahora pasamos de S y R=0 a S=1 y R=0



Memoria

- Se puede construir con un flip-flop una memoria de 1 bit.
- Se llama biestable porque el circuito posee sólo 2 estados posibles de funcionamiento, se queda en cada uno de ellos, salvo que las entradas provoquen un cambio.



Secuenciales - Clasificación

- Según la manera en que las salidas respondan a las señales lógicas presentes en la entrada, los biestables se clasifican en:
 - SR
 - J-K
 - D
 - T



Secuenciales – Clasificación(2)

- Respecto del instante en que pueden cambiar dichas salidas, pueden ser:
 - **Asincrónicos**: cuando en la entrada se establece una combinación, las salidas cambiarán
 - **Sincrónicos**: la presencia de una entrada especial, determina “cuando” cambian las salidas acorde a las entradas

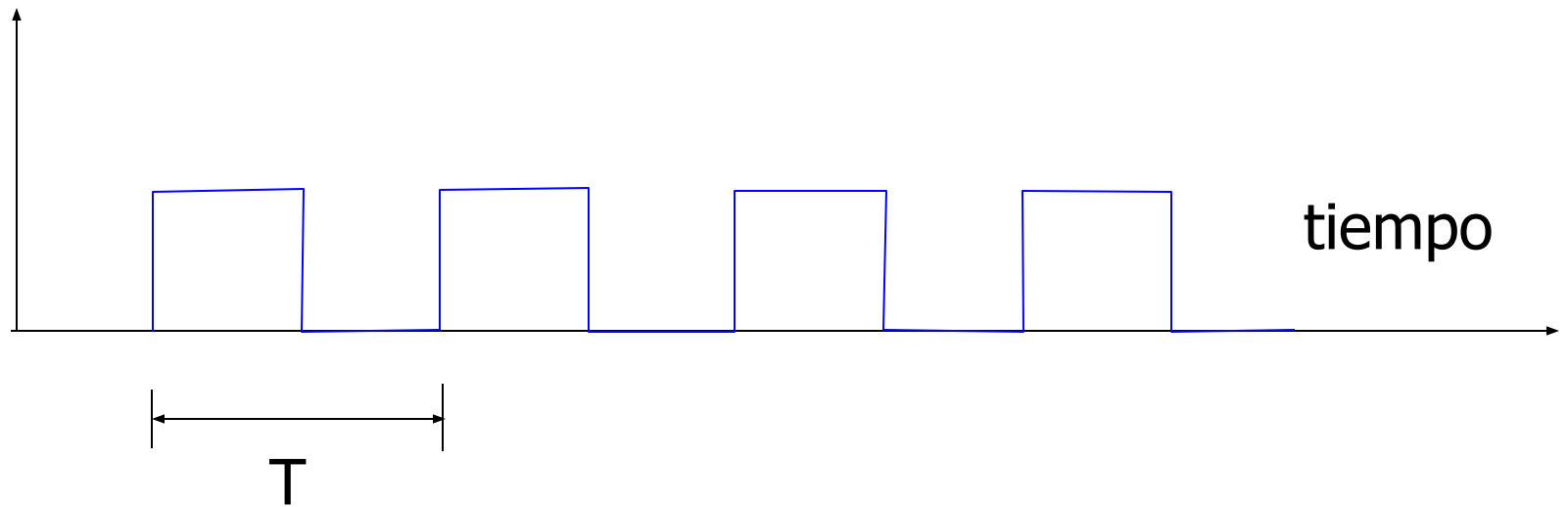


Reloj: “señal especial”

- El orden en que ocurren los sucesos es importante.
- A veces los sucesos deben ocurrir simultáneamente.
- Reloj: es una señal de tiempo precisa que determina cuando se producen eventos.

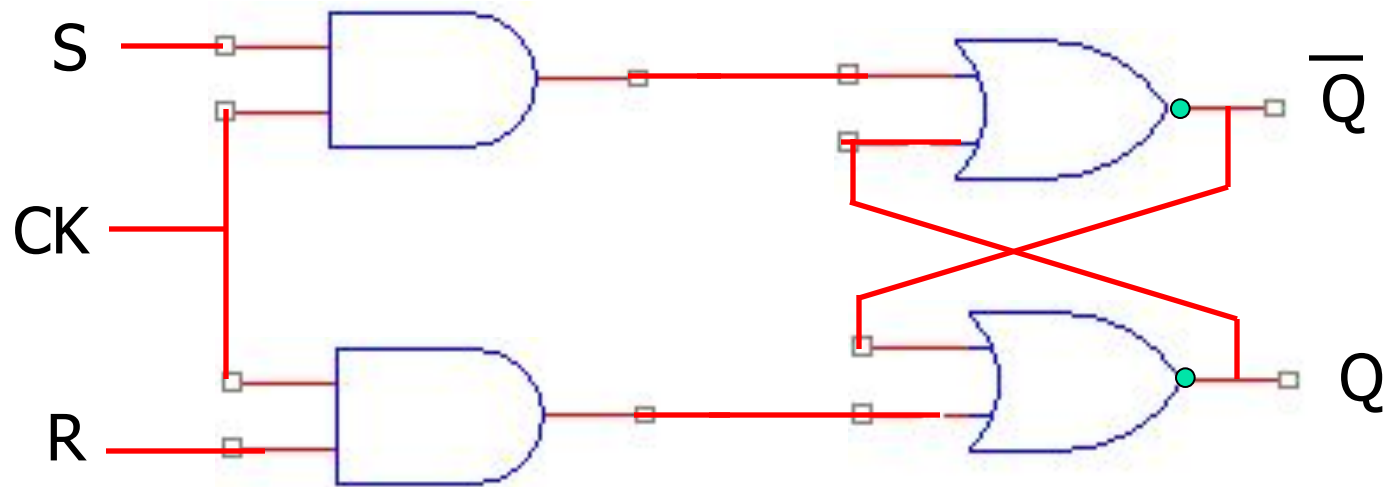


Reloj (Clock) (CLK)



Cada tiempo T , la señal se repite

Flip-Flop SR sincrónico



- S y R son las entradas que tendrán efecto cuando CK tome el valor 1.

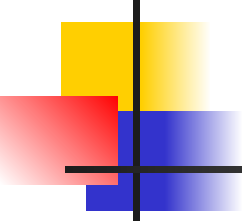
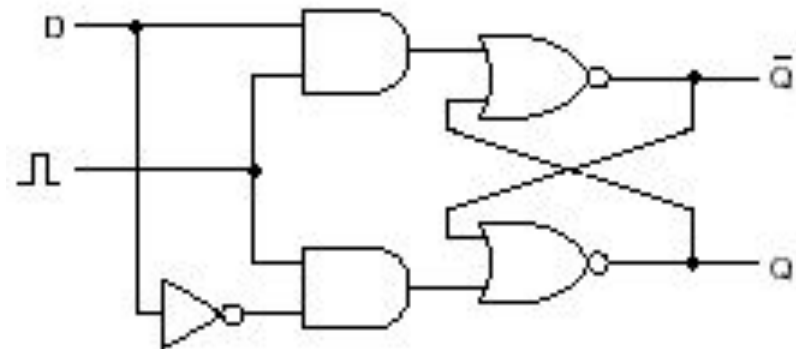
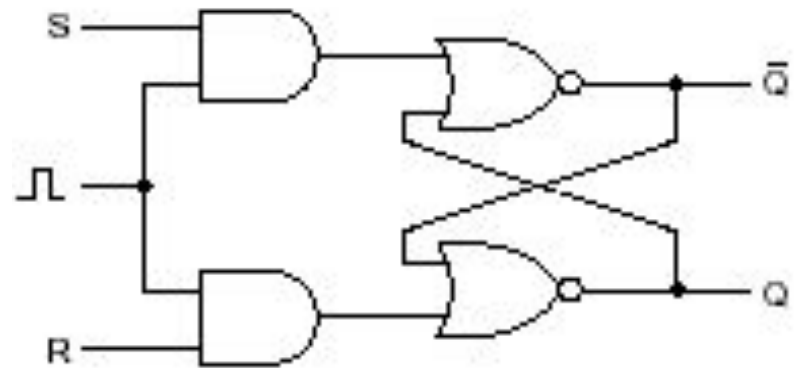


Tabla de comportamiento: SR sincrónico

CK	S	R	Q_{n+1}
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Prohibido
0	x	x	Q_n

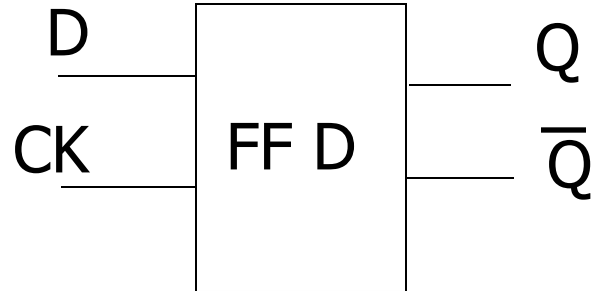
Flip-Flop D

- En el FF SR hay que aplicar 2 entradas diferentes para cambiar de estado.
- El FF D permite aplicar una sola entrada para cambiar la salida.





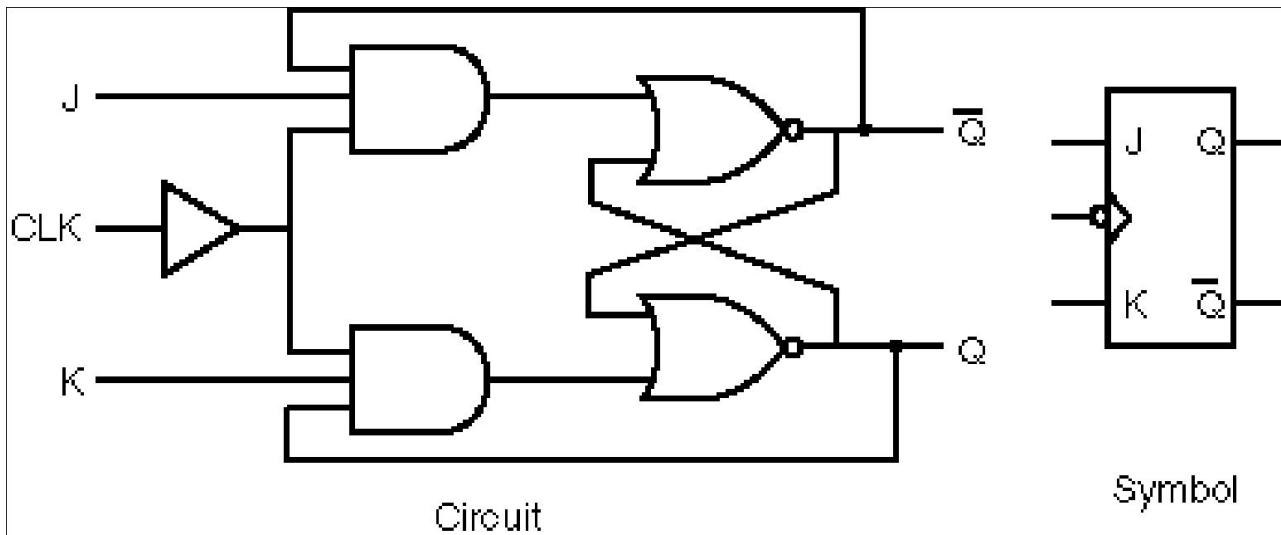
Flip-Flop D



D	Q_{n+1}
0	0
1	1

con CK=1

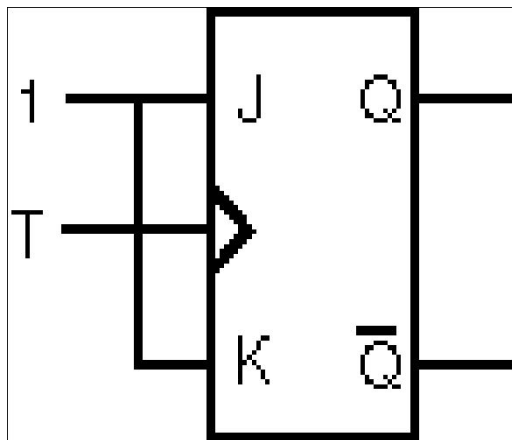
Flip Flop J-K



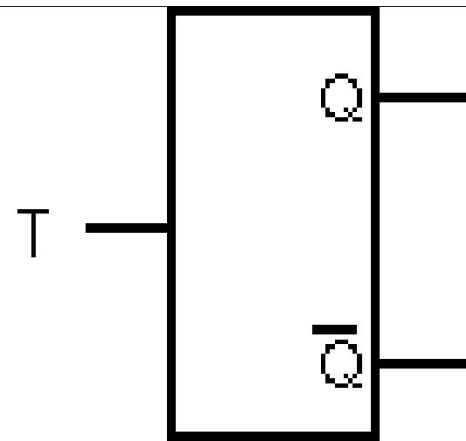
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

Flip Flop T

- La salida Q cambiará de 0 a 1 o 1 a 0 en cada pulso de la entrada T.



Circuit



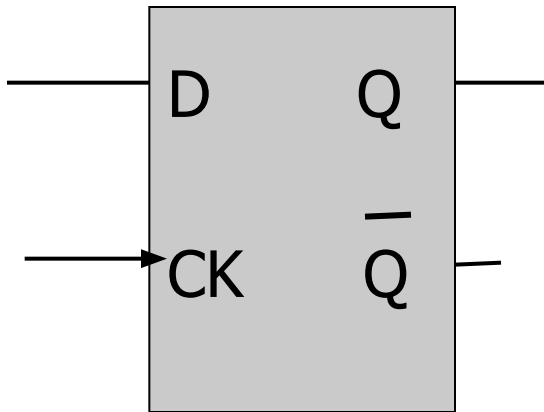
Symbol



Recordando un bit

- Con una señal (CK) se copia el valor de D en Q
- Sin esa señal, el valor de Q permanece igual

Puedo recordar un Bit

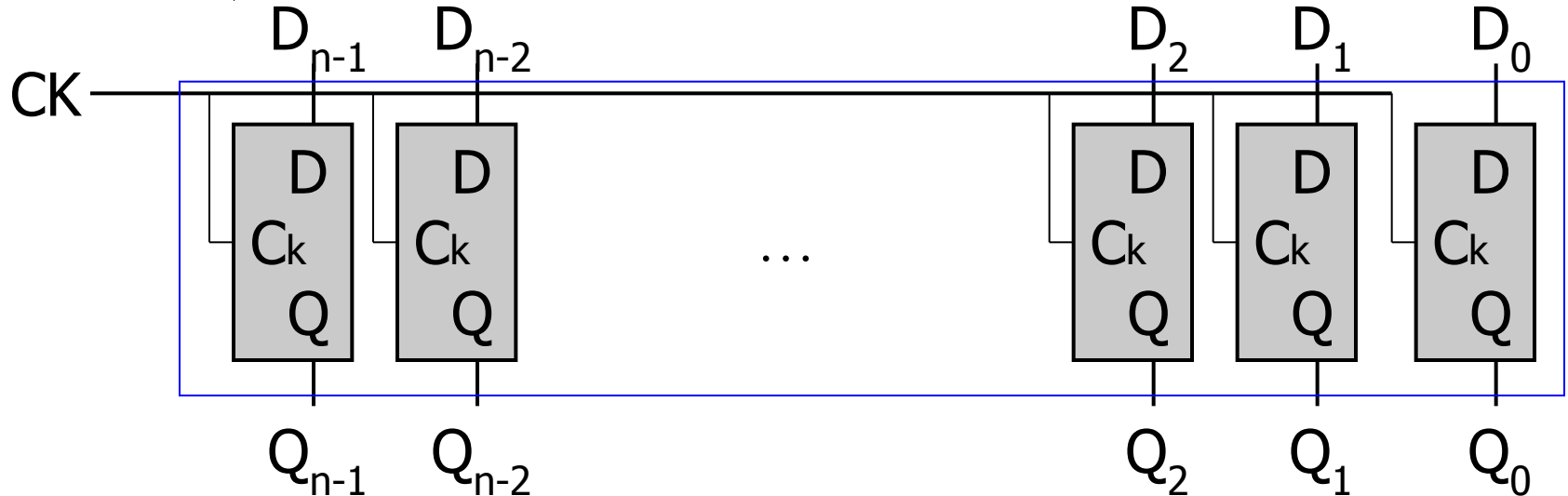


CK	D	Q
0	0	q
0	1	q
1	0	0
1	1	1

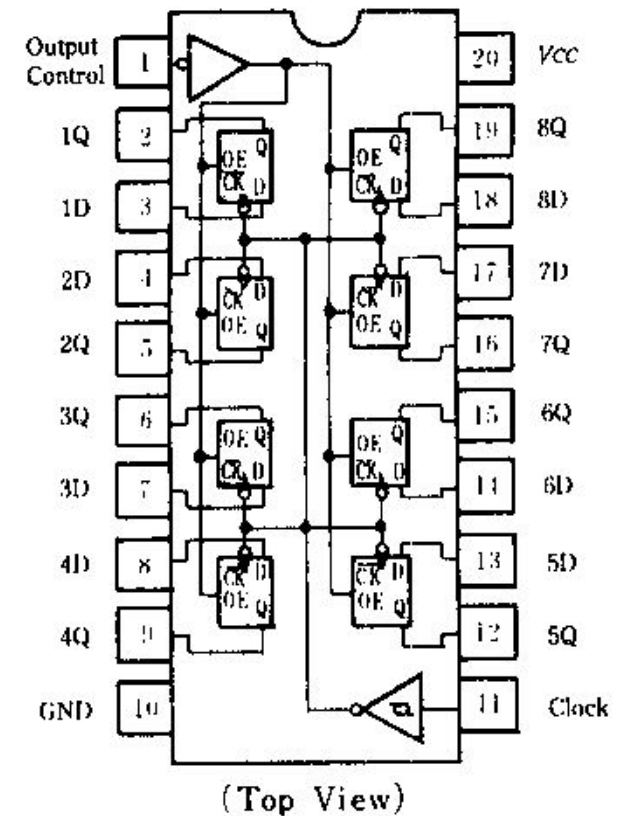
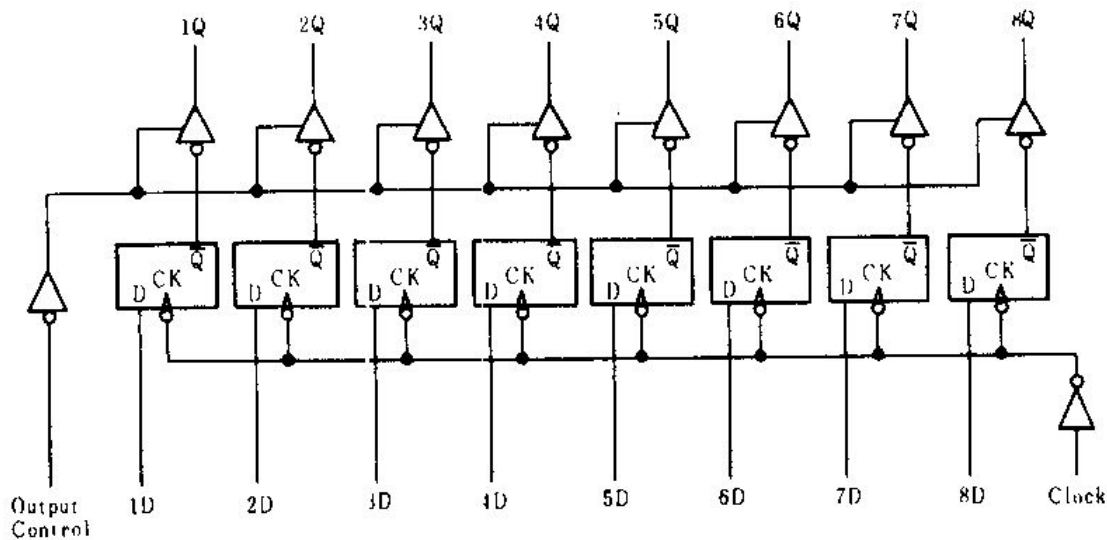
Recordando n bits

- Si CK actúa sobre n bits simultáneamente

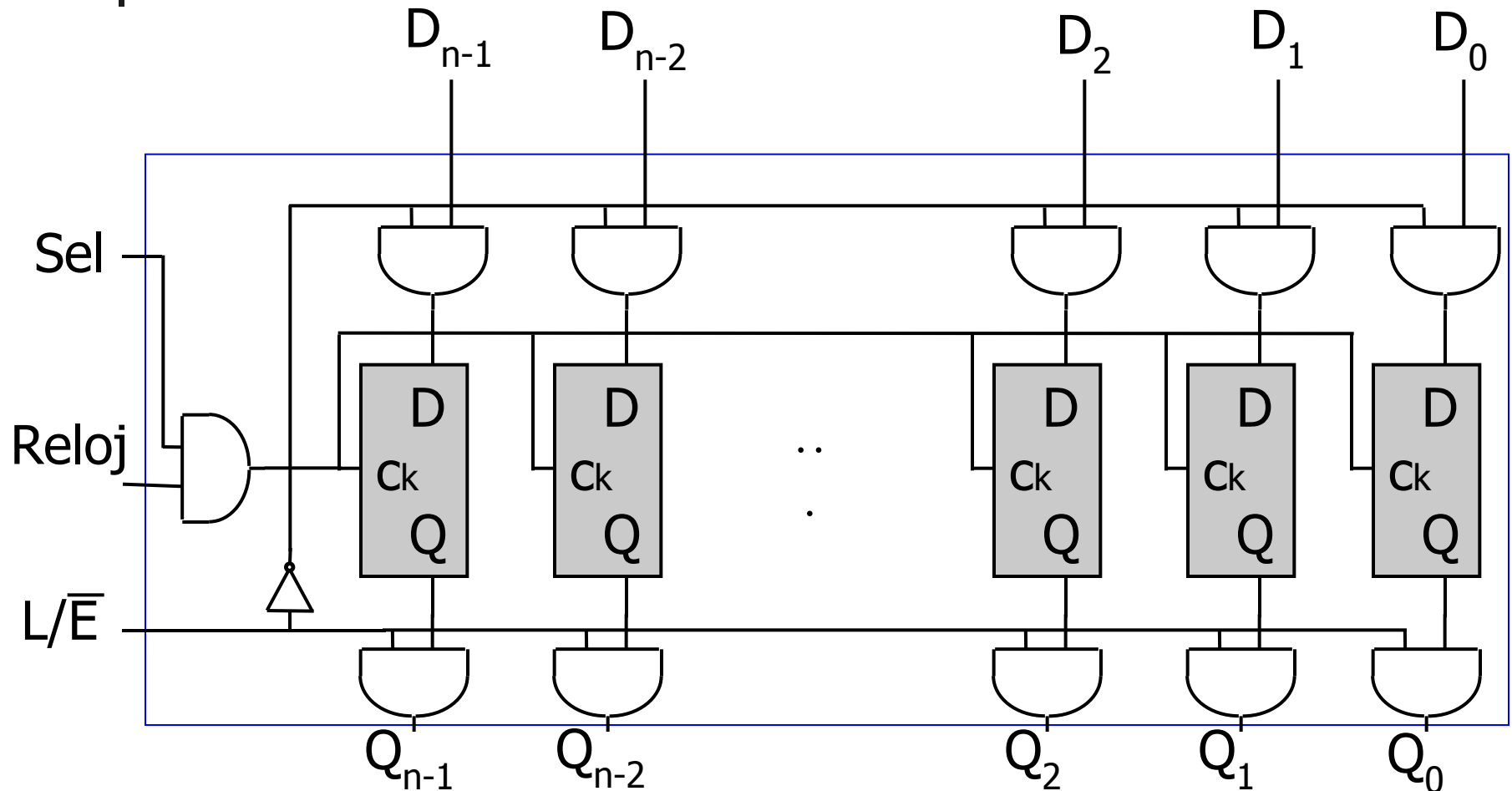
Registro n bits



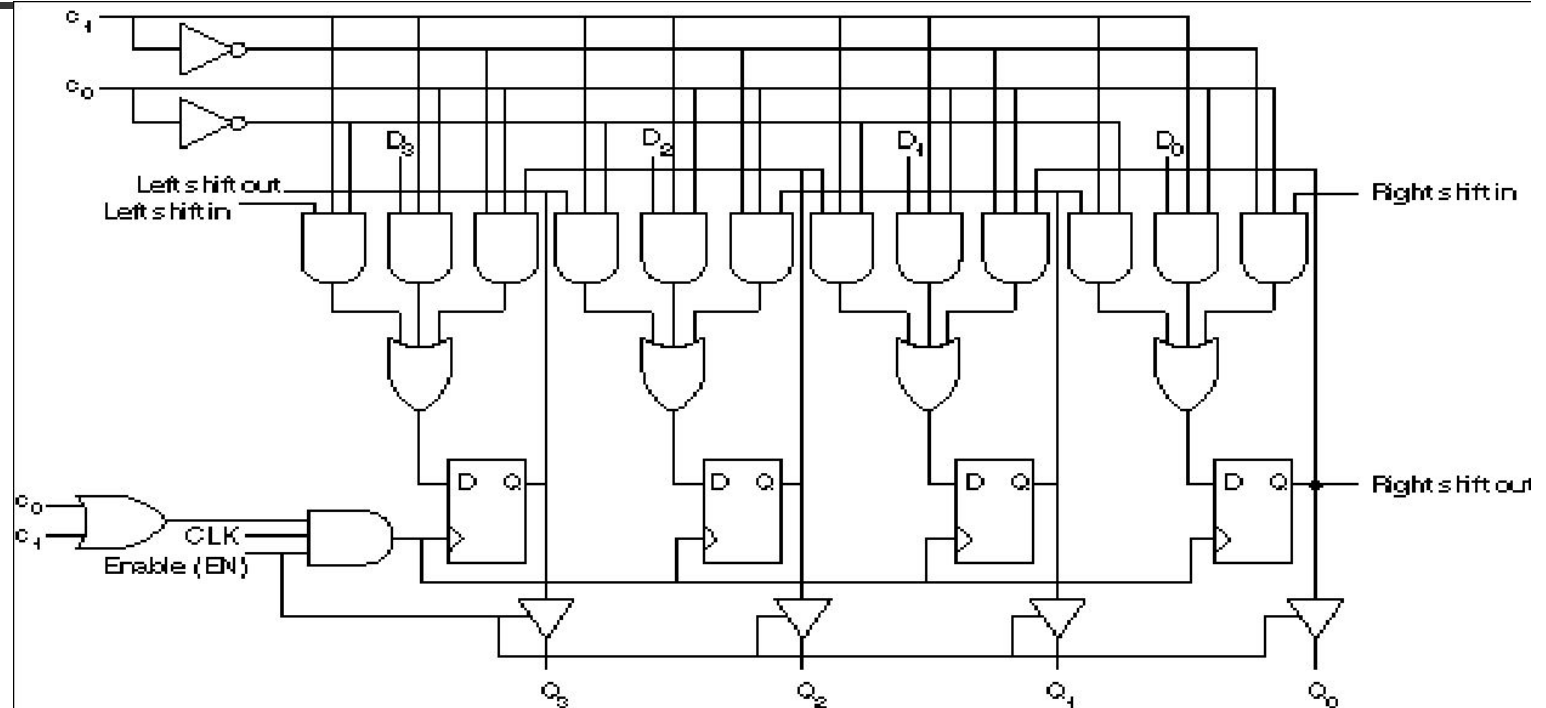
Chip con 8 FF-D (74LS374)



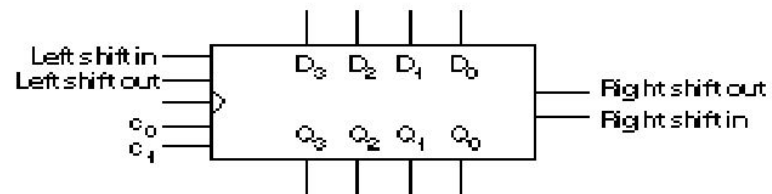
Selección y operaciones



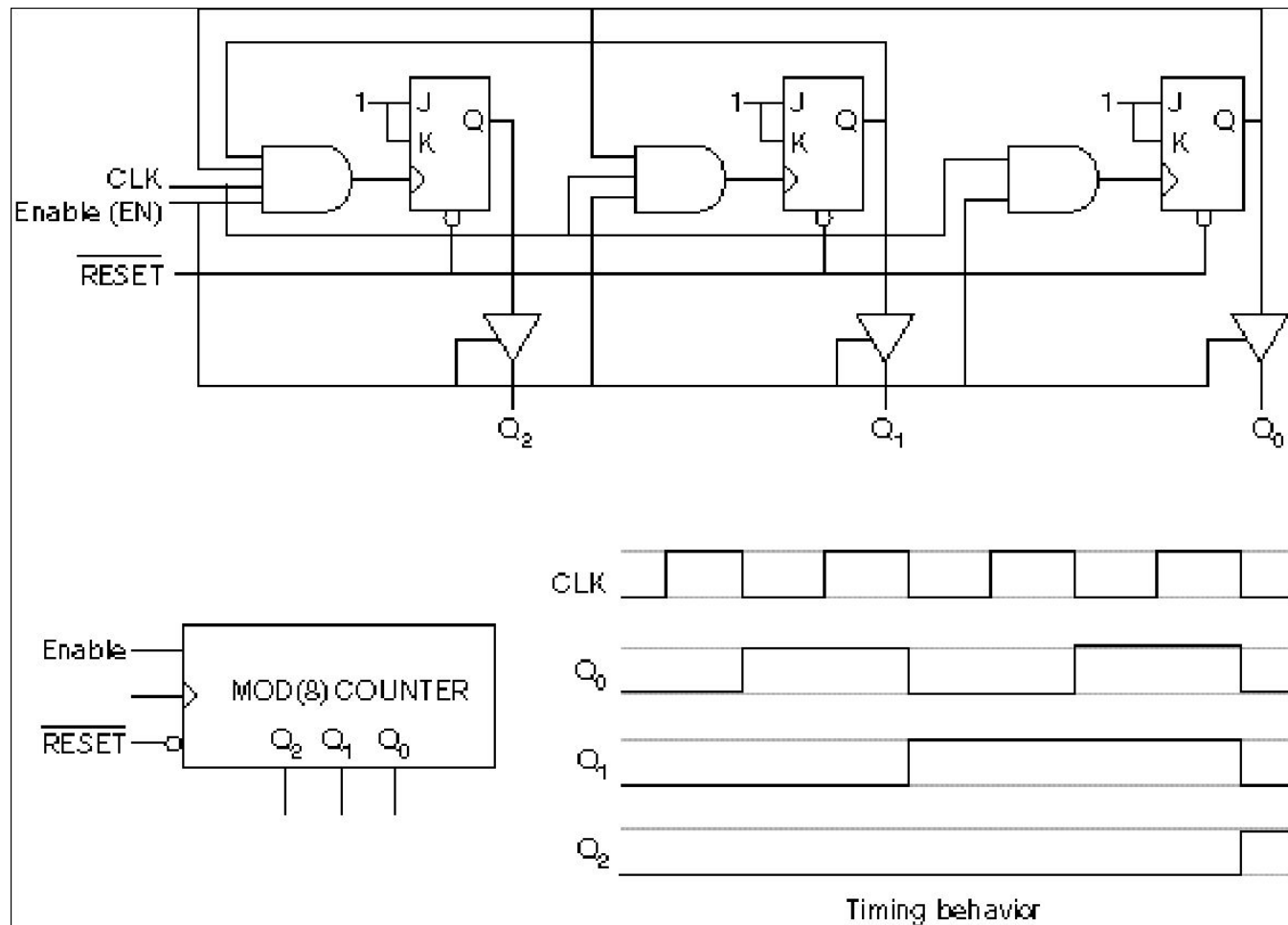
Registro con desplazamiento



Control		Function
c_1	c_0	
0	0	No change
0	1	Shift left
1	0	Shift right
1	1	Parallel load



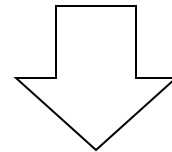
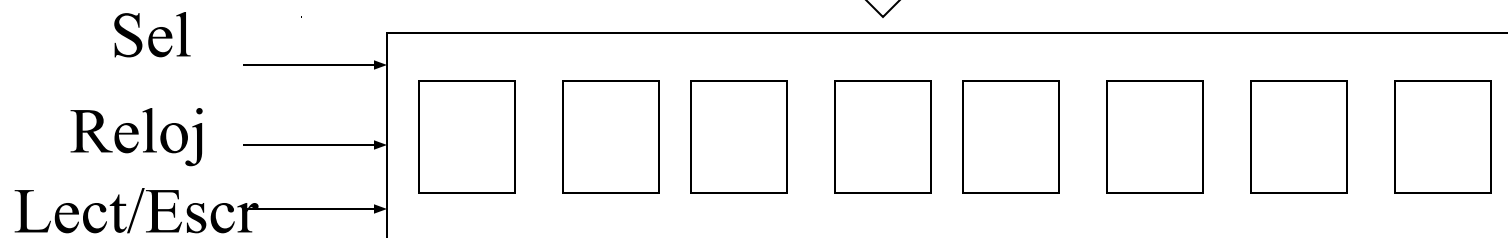
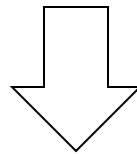
Contador módulo 8





Un Registro

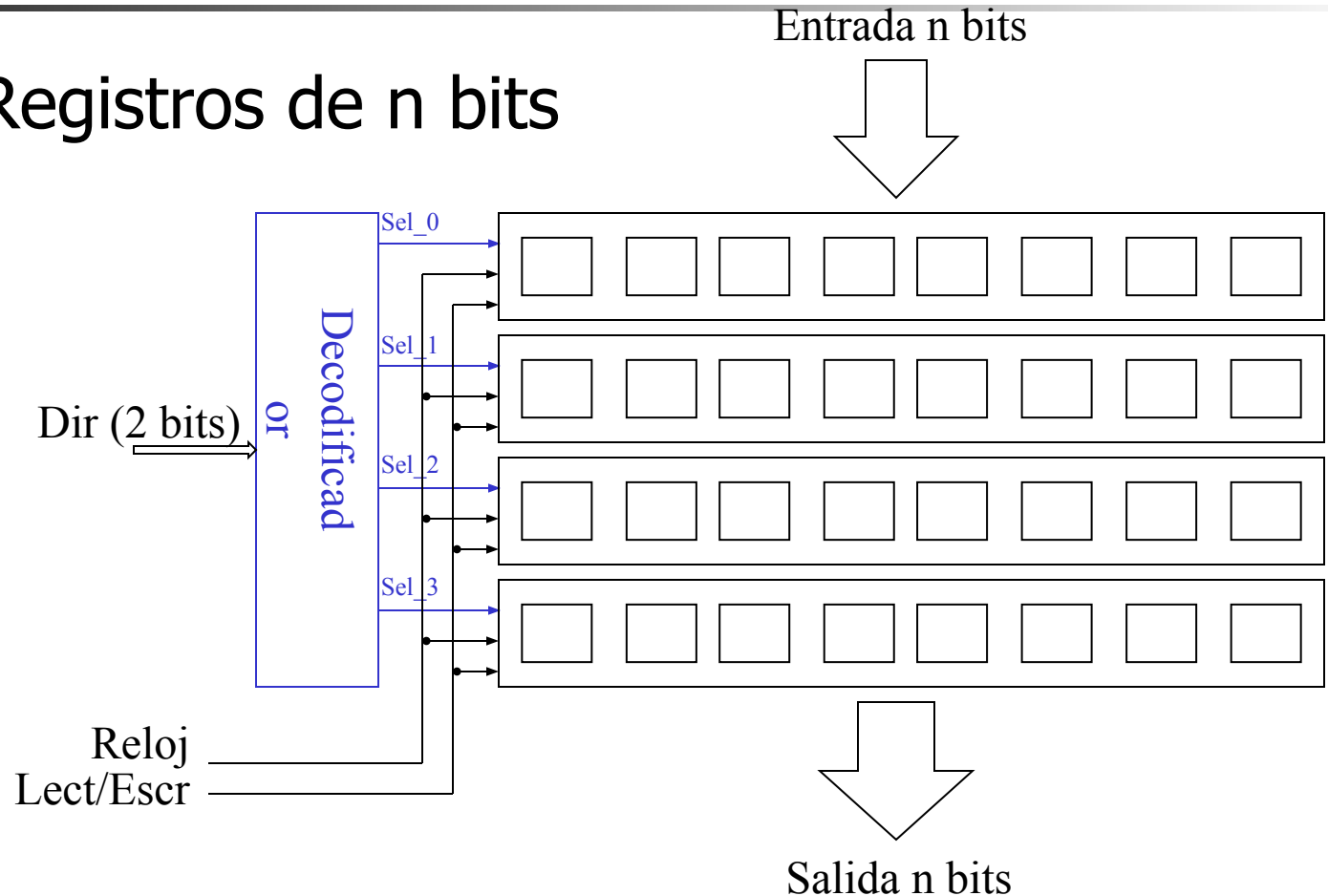
Entrada n bits



Salida n bits

Varios Registros

4 Registros de n bits





mayor información ...

- Operaciones Lógicas
 - Apunte 3 de Cátedra
- Circuitos Secuenciales
 - Apunte 5 de Cátedra
- Apéndice A: Lógica digital (A.3., A.4.)
 - Stallings, W., 5º Edición.